

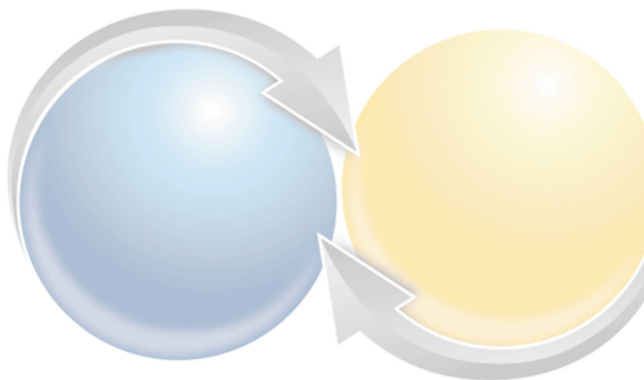


Livelink WCM Server

Integration Manual

This manual describes how to integrate your WCM system with a Livelink system. It provides information on the following topics:

- setting up access to the Livelink user administration for the WCM system
- using Livelink functions in a website
- managing Livelink contents in the Content client
- using the Livelink search engine for a website



Copyright 2006 by Open Text Corporation.

The copyright to these materials and any accompanying software is owned, without reservation, by Open Text Corporation. These materials and any accompanying software may not be copied in whole or part without the express, written permission of Open Text Corporation.

Open Text Corporation is the owner of the trademarks Open Text, 'Great Minds Working Together', Livelink, and MeetingZone among others. This list is not exhaustive. All other products or company names are used for identification purposes only, and are trademarks of their respective owners. All rights reserved.

Open Text Corporation provides certain warranties and limitations in connection with the software that this document describes. For information about these warranties and limitations, refer to the license agreement entered into between the licensee and Open Text Corporation.

Contacting Us

Open Text Corporation
Corporate Headquarters
275 Frank Tompa Drive,
Waterloo, Ontario, Canada
N2L 0A1

(519) 888-7111

If you subscribe to our Customer Assistance Program or would like more information about the support program, write to Open Text Corporation's Customer Support at support@opentext.com or telephone (800) 540-7292 or (519) 888-9933. Our support hours are Monday through Friday, 8:30 a.m. to 8 p.m. (EST).

If you have comments or suggestions regarding this documentation, write to the Open Text Corporation Publications Group at documentation@opentext.com.

For more information about Open Text Corporation's products and services, visit our home page at <http://www.opentext.com>.

© 2006 IXOS SOFTWARE AG

Werner-v.-Siemens-Ring 20
85630 Grasbrunn, Germany

Tel.: +49 (89) 4629-0

Fax: +49 (89) 4629-1199

eMail: office@ixos.de

Internet: <http://www.ixos.com>

All rights reserved, including those regarding reproduction, copying or other use or communication of the contents of this document or parts thereof. No part of this publication may be reproduced, transmitted to third parties, processed using electronic retrieval systems, copied, distributed or used for public demonstration in any form without the written consent of IXOS SOFTWARE AG. We reserve the right to update or modify the contents. Any and all information that appears within illustrations of screenshots is

provided coincidentally to better demonstrate the functioning of the software. IXOS SOFTWARE AG hereby declares that this information reflects no statistics of nor has any validity for any existing company. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>) and software developed by the Apache Software Foundation (<http://www.apache.org/>).

Trademarks IXOS: IXOS SOFTWARE AG.

SAP[®], R/3[®] and SAP ArchiveLink[®] are registered trademarks of SAP AG.

Microsoft[®], Microsoft Windows NT[®] and the names of further Microsoft products are registered trademarks of Microsoft Corporation.

Acrobat Reader Copyright © 1987 Adobe Systems Incorporated. All rights reserved. Adobe and Acrobat are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.

Siebel[®] is a registered trademark by Siebel Systems, Inc.

Other product names are used only to identify the products and they may be registered trademarks of the relevant manufacturers.

Copyright © 2006 Gauss Interprise AG Hamburg, Gauss Interprise, Inc. Irvine, California. All rights reserved worldwide.

This document and the related software are property of Gauss Interprise AG or its suppliers and protected by copyright and other laws. They are distributed under licenses restricting their use, copying, distribution, and decompilation. Neither receipt nor possession of this document confers or transfers any right to reproduce or disclose any part of the contents hereof. No part of this document may be reproduced in any form by any means without prior written authorization of Gauss Interprise AG or Gauss Interprise, Inc.

Moreover, the regulations of the software license agreement apply to this documentation.

All brand names and trademarks mentioned are the property of their respective owners.

<http://www.opentext.com/bridging/gauss.html>

Program version: Livelink Web Content Management Server[™] (Content Server) 9.5.1

Document version: En-01

Publication date: May 2006

Table of Contents

List of Figures		6
List of Tables		9
Chapter 1	Introduction	11
	Integration Procedure	13
	About this Manual	14
	Typographic Conventions	15
Chapter 2	Common User Administration	19
	Setting up a Common User Administration	19
	Managing Principals in the Admin Client	26
	Managing Principals in Livelink	54
Chapter 3	Livelink Functions in a Website	81
	Preparations	82
	Configuring Multiple Livelink Systems	86
	Overview of API Functions	91
	Application Examples	95
Chapter 4	Livelink Contents in the Content Client	131
	Adding Relators	133
	Editing Relators	149
	Deleting a Relator	163

Chapter 5	Livelihood Search in the WCM System	167
	Technical Concepts	168
	Configuring the Livelihood Search for the WCM System	171
	Permissions for Searching	203
	Using the Search	204
	Command Line Tools	205
	Problems and Solutions	215
Glossary		219
Index		227

List of Figures

Fig. 1 –	The <i>Module Administration</i> section on the Livelink Administration page	21
Fig. 2 –	Specifying the user administration when installing the WCM system	22
Fig. 3 –	Logging in to the Admin client	27
Fig. 4 –	User administration items in the Admin client	30
Fig. 5 –	Finding principals in the Admin client	32
Fig. 6 –	Stored users in the Admin client	33
Fig. 7 –	Adding a user with the Admin client	35
Fig. 8 –	Modifying settings of a user in the Admin client	36
Fig. 9 –	Assignments of a user in the Admin client	37
Fig. 10 –	Stored groups in the Admin client	41
Fig. 11 –	Adding a group in the Admin client	42
Fig. 12 –	Assignments of a group in the Admin client	44
Fig. 13 –	Available functional areas	48
Fig. 14 –	Adding a functional area	49
Fig. 15 –	Settings of functional areas	50
Fig. 16 –	Website assignments	53
Fig. 17 –	Logging in to the Livelink system	54
Fig. 18 –	Finding principals in Livelink	58
Fig. 19 –	Users found in Livelink	59
Fig. 20 –	Users stored in Livelink	60
Fig. 21 –	Adding a user in Livelink	62
Fig. 22 –	Editing user information in Livelink	65
Fig. 23 –	Groups stored in Livelink	73
Fig. 24 –	Adding a group in Livelink	74
Fig. 25 –	Editing group information in Livelink	74

Fig. 26 –	Adding members to a group in Livelink	76
Fig. 27 –	Group members in Livelink	77
Fig. 28 –	Setting a user as the group leader in Livelink	78
Fig. 29 –	The <i>Module Administration</i> section on the Livelink Administration page	83
Fig. 30 –	Website extension for Livelink	84
Fig. 31 –	Assigning the Livelink pool to the master Content server	85
Fig. 32 –	Selecting a new run level	87
Fig. 33 –	Specifying the Livelink system	135
Fig. 34 –	Selecting the Livelink object when adding a relator	136
Fig. 35 –	Specifying the version of the Livelink object	137
Fig. 36 –	Entering comments about the relator	138
Fig. 37 –	Specifying the Livelink system	140
Fig. 38 –	Selecting the Livelink folder	141
Fig. 39 –	Specifying the version of the Livelink folder relator	142
Fig. 40 –	Specifying title, file, and object type	145
Fig. 41 –	Specifying the Livelink system	147
Fig. 42 –	Selecting the Livelink folder	148
Fig. 43 –	Object filter in the Content client	151
Fig. 44 –	Filter results	151
Fig. 45 –	The Livelink metadata dialog box	157
Fig. 46 –	The category “Training Materials” in the Livelink system	161
Fig. 47 –	Interaction of components in the Livelink search	170
Fig. 48 –	Search servers in the WCM installation program	174
Fig. 49 –	Installing Livelink Search servers	175
Fig. 50 –	The <i>Search Administration</i> section on the Livelink Administration page	179
Fig. 51 –	Creating an XML Activator Producer Data Source	180
Fig. 52 –	Specifying the Livelink parameters for an internal collection	183
Fig. 53 –	The <i>Start Options</i> section on the <i>Specific</i> tab	186

Fig. 54 –	Search servers in the WCM installation program	190
Fig. 55 –	Installing Livelink Search servers	191
Fig. 56 –	The <i>Search Administration</i> section on the Livelink Administration page	194
Fig. 57 –	Creating an XML Activator Producer Data Source	195
Fig. 58 –	Specifying the Livelink parameters for an internal collection	198
Fig. 59 –	The <i>Start Options</i> section on the <i>Specific</i> tab	201

List of Tables

Table 1 –	Notes on common user administration with the Admin client	28
Table 2 –	Items and functions of the user administration in the Admin client	30
Table 3 –	Notes on managing users with Livelink	55
Table 4 –	Available functional areas	67
Table 5 –	Actions and required default object rights	70
Table 6 –	Elements of the livelink-systems.xml	88
Table 7 –	Livelink object types and their equivalents in the WCM Java API	92
Table 8 –	Livelink users and groups and their equivalents in the WCM Java API	93
Table 9 –	WCM tags for the most common Livelink metadata	159
Table 10 –	General parameters of the command line tools	208
Table 11 –	Parameters of the command line tool AddDocument	209
Table 12 –	Parameters of the command line tool DeleteDocument	209
Table 13 –	Parameters of the command line tool AddDirectory	210
Table 14 –	Parameters of the command line tool DeleteDirectory	211
Table 15 –	Parameters of the command line tool ListSources	212
Table 16 –	Parameter of the command line tool RemoveCollection	213
Table 17 –	Parameters of the command line tool Query	214

CHAPTER 1

Introduction

Livelihood Web Content Management Server™ (Livelihood WCM Server for short) is a user-friendly, highly scalable, and adaptable content management system for setting up and managing sophisticated and high-performance intranet and Internet solutions.

Thanks to the integration with Open Text Livelihood Enterprise Server™ (Livelihood for short), a comprehensive enterprise content management solution has been developed which covers all aspects of collaboration for creating, managing, and publishing content in intranets, extranets, and the Internet.



The main features of the integration are described briefly below.

Common user administration

By integrating the web content management system (WCM system for short) with Livelink, you can use the Livelink user information for both systems. The user data will be stored in Livelink. For managing principals (users, groups, and roles), you can either use the Livelink user administration or the Admin client of the WCM system. Managing users belongs to the tasks of system administrators. If you are a system administrator, read about this topic in chapter 2 “Common User Administration”.

Publishing Livelink objects in a website

You can use JSP pages to easily and conveniently publish Livelink objects, such as folders, documents, discussions, and channels, in a WCM-managed website. For this purpose, you require programming know-how. If you have the task to program JSP pages, read about this topic in chapter 3 “Livelink Functions in a Website”.

Another way of publishing Livelink objects in a website is to add Livelink objects as WCM objects to a website. These WCM objects are called *relators*. Relators enable you to integrate Livelink contents in a website without the need of programming. If you are an editor and want to publish Livelink contents in a website in an easy and comfortable way, read chapter 4 “Livelink Contents in the Content Client”.

Using the Livelink search for a website

The Search Server Connector for Livelink enables you to use the Livelink search engine for a WCM-managed website. The configuration and administration of search engines belong to the tasks of system administrators. If you are a system administrator, read about this topic in chapter 5 “Livelink Search in the WCM System”.

Integration Procedure

In order to use the WCM system together with Livelink, the following steps are required. It is presumed that Livelink is already installed.

1. Install the WCM system with a Livelink user administration (system administrator).

See Livelink WCM Server Installation Manual

2. Set up a website with the Admin client of the WCM system (system administrator).

See online help of the Admin client

3. Edit principals, i.e. users and groups (system administrator)

The principals must be able to access the website created in the previous step. They also need the appropriate functional areas and object rights.

See chapter 2 “Common User Administration”

4. Optional: Create JSP pages in order to use Livelink objects, such as news and discussions, in a website (web designer).

See chapter 3 “Livelink Functions in a Website”

5. Optional: Add relators (editors).

The editors can continue to use Livelink as their authoring environment.

See chapter 4 “Livelink Contents in the Content Client”

About this Manual

This manual describes the different areas that the Livelink integration covers. The description is based on Livelink Enterprise Server 9.2.1. The manual is intended for

- editors, web designers, and programmers who set up new websites or manage and extend existing websites with Livelink WCM Server in order to publish documents and other Livelink content in an intranet, extranet, or Internet presence.
- administrators managing the system environment for the websites and integrating the WCM system with an existing Livelink installation.

In addition to this Integration Manual, the following sources provide information:

- **Livelink WCM Server Installation Manual:** This document describes how to install the WCM system and configure web servers and application servers.
- **Livelink WCM Server Administrator Manual:** This document explains how to configure and administer WCM systems. Moreover, it contains a detailed description of the technical concepts of Livelink WCM Server.
- **Content Client User Manual:** This manual provides detailed instructions on all tasks involved in editorial work on websites managed with Livelink WCM Server.
- **WCM Java API Programmer's Manual:** This document contains information on interfaces, classes, and methods of the Java programming interface (WCM Java API), which enables you to use the functionality of the WCM servers via external programs.

- Portal Manager API Programmer's Manual: This manual contains detailed information on developing dynamic and personalized websites on the basis of the Portal Manager API.
- Content Miner Manual and Search Server Connector for Lucene Manual: These documents describe concepts and management of a Search server.

For detailed information about concepts and use of Livelink, refer to the Livelink documentation.

Typographic Conventions

The following conventions are used in the text to draw attention to program elements, etc.:

Element	Font or symbol	Examples
Program interface such as menu commands, windows, dialog boxes, field and button names	<i>Menu → Entry</i>	<i>File → Create</i>
Paths to directories, file and directory names	Drive:\Directory\File name	D:\WCM\admin.bat
Quotations from program code or configuration files	Code quotations	<code><head> <title>heading </title> </head></code>
Variables, i.e. placeholders for specific elements	{variable}	{WCM installation directory}

Important **information** and **warnings** are enclosed in gray boxes. Make sure to read such information to avoid losing data or making errors when using and managing WCM systems.

CHAPTER 2

Common User Administration

By integrating the WCM system with Livelink, you can use the Livelink user information for both systems. The user data will be stored in Livelink. For managing principals (users, groups, and roles), you can either use the Livelink user administration or the Admin client of the WCM system.

This chapter deals with the following topics:

- setting up a common user administration, see the following section
- editing principals in the Admin client, see section “Managing Principals in the Admin Client” starting on page 26
- editing principals in the Livelink user administration, see section “Managing Principals in Livelink” starting on page 54

Setting up a Common User Administration

In order to manage Livelink and WCM users in a common data storage, you have to perform two installation steps:

1. Install the Livelink module “WCM User Administration” (see the following section).
2. Select a Livelink user administration when installing the WCM system (see section “Selecting the Livelink User Administration” on page 22).

Installing the WCM User Administration Module

Notes

- Installing the module is only required if you want to use the option *Livelihood* for the user administration when installing your WCM system.
- If the Livelihood user information is synchronized with an LDAP directory service, we recommend that you select the *LDAP* option for the user administration when installing the WCM system. In this case, you have to use the Admin client for editing WCM-specific principal settings, as the Livelihood user administration does not provide the *WCM* tab.
- The module must be installed **prior** to installing the WCM system.

Procedure

1. The WCM User Administration module is supplied as file **wcmadm.zip**. This file is located in the `\\livelihood\` directory on the WCM CD. Extract the file to the directory **{Livelihood installation directory}\staging**.
2. Open the Livelihood Administration page.

The URL is of type
<http://livelihoodserver.company.example/Livelihood/livelihood.exe?func=admin.index>.
3. Log in to the Livelihood system as the Admin user and go to the *Module Administration* section.

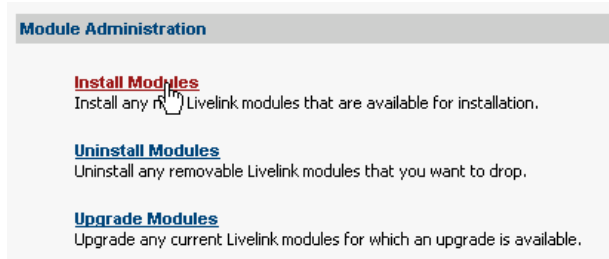


Fig. 1 – The *Module Administration* section on the Livelink Administration page

4. Click the *Install Modules* link.
5. On the Install Modules page, select the check box of the WCM User Administration module and then click the *Install* button.

For further information on installing modules, refer to the Livelink online help.

Uninstalling the module

After uninstalling the module, the WCM system is no longer available, as the users cannot log in. For this reason, you should **first** uninstall the WCM system and then the module.

Click the *Uninstall Modules* link in the *Module Administration* section on the Livelink Administration page. For further information on uninstalling modules, refer to the Livelink online help.

Selecting the Livelink User Administration

In the *Admin server* dialog box of the WCM installation program, you specify whether the Administration server stores the user data in a relational database management system, in an LDAP directory service, or in Livelink.

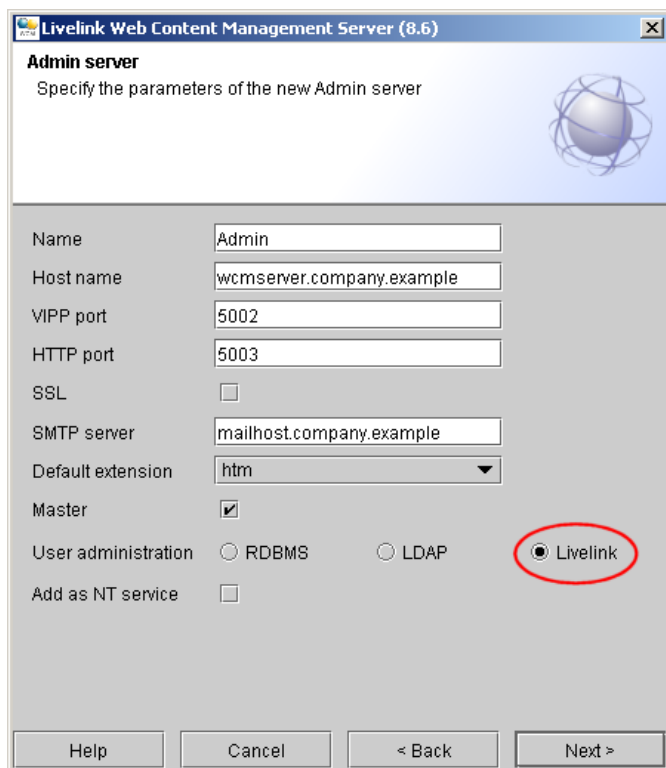


Fig. 2 – Specifying the user administration when installing the WCM system

Click the *Livelink* radio button. In the next dialog box, you enter the parameters for the connection to the Livelink system.

For detailed information on installing the WCM system, refer to the Livelink WCM Server Installation Manual.

After the Installation

After setting up a common user administration for the WCM system and Livelink, all Livelink users and groups automatically have access to the WCM system. However, they cannot access any websites yet. After the installation, you must enable website access for the principals and thus make it possible for them to work in the WCM system.

A clear access strategy allows you to control object access without great effort. Organizing users in groups is a precondition for an access strategy. Assigning rights to individual users should be the exception.

Proceed as follows:

1. Assign groups to websites.

The assignment of groups to a website enables the users to access the website objects with the Content client.

See section “Website Assignments” on page 52.

2. Assign functional areas to groups.

Functional areas allow you to control which types of objects the users may create, check in, and check out as well as which views, dialog boxes, and functions are available to them in the Content client.

See section “Assigning Functional Areas” on page 51.

3. Assign default object rights to groups.

For each principal, you can define default access rights for website objects. The object rights determine which actions (“Read”, “Change metadata”, “Release”, etc.) a user can perform by default on a WCM object to which the user is assigned.

See section “Modifying the Assignments of a Group or Role in the Admin Client” on page 43.

4. Repeat steps 1 to 3 for individual users, if required.

You can either use the Admin client or the Livelink user administration to make the user assignments.

Admin client:

- assigning websites, see section “Website Assignments” on page 52
- assigning functional areas, see section “Assigning Functional Areas” on page 51
- assigning default object rights, see section “Modifying User Assignments in the Admin Client” on page 37

Livelink: All assignments are made in the user settings, see section “Settings on the WCM tab” on page 66.

5. Assign administration rights for the WCM system.

Access to the administration and configuration of the WCM system should be reserved to selected users. You can grant administration rights to users, groups, and roles.

See sections “Modifying User Assignments in the Admin Client” on page 37 and “Modifying the Assignments of a Group or Role in the Admin Client” on page 43.

For detailed information on how to develop an access strategy for the WCM system, refer to the Livelink WCM Server Administrator Manual (chapter “Concepts”).

Accessing Several Livelink Servers

Currently, the WCM User Administration module does not support access to several Livelink servers from the same Admin server (as it is possible with LDAP servers). Each Admin server can only access the user administration of exactly one Livelink server.

For information on managing objects from several Livelink systems in one WCM system, refer to section “Configuring Multiple Livelink Systems” on page 86.

Accessing Several WCM Systems

Currently, the WCM User Administration module does not support the integration of more than one WCM system with a Livelink server's user administration.

Nevertheless, you can configure the WCM systems in such a way that they access the user administration of the same Livelink server. In this case, please consider the following:

- The responsible Admin server automatically updates the information on the available websites and functional areas in Livelink each time the information is changed in one of the connected WCM systems. This means that the information on the *WCM* tab in Livelink always reflects the state of the last WCM system that has been changed.
- All changes to the principals of one WCM system also apply to all other WCM systems connected to the user administration of the same Livelink server. Thus it is not possible, for example, to grant a user a specific default object right for only one WCM system.

Managing Principals in the Admin Client

Notes:

In order to add, edit, and delete principals in the Admin client, you need the administration right “Create, modify, delete principal”. In order to modify the administration rights of a principal, you also need the right “Change administration rights”.

Users with appropriate administration rights in the WCM system can make changes to the user information which also affect Livelink. These users do not require any privileges in the Livelink system for this.

You can manage all users who are to access the WCM system and the Livelink system with the Admin client. To group users according to their tasks and the company's organizational structure and to assign special access rights, you can define groups and roles and assign functional areas.

Starting the Admin client

If the Admin client is installed on your computer, use the batch file **adminClient.bat** (Windows) or **adminClient.sh** (UNIX) to launch the client. These files are located in the directory **{WCM installation directory}\admin**. To start the web application of an Admin client, open your browser and go to the URL of the web application's start page (e.g. <http://wcmserver.company.example/wcmadminclient/index.html>).

Immediately after the installation of the WCM system, the only user with administration rights is the one you entered during the installation.

Logging in and out of the Admin client

After you have started the Admin client, the Login dialog box opens in the language selected for installing the WCM system.



Fig. 3 – Logging in to the Admin client

Enter your user ID and the corresponding password. If this is the first time you start the Admin client after the installation of the WCM system, enter the user ID and password of the user who was configured as the administrator during the installation process.

When you log in to the WCM system for the first time, you are asked to change your password.

Note: Both user ID and password are case sensitive. Correct use of uppercase and lowercase letters is essential. For Livelink WCM Server, “administrator” and “Administrator” are two different users.

To exit the Admin client, choose *Quit* on the *Program* menu. The connection to the Admin server will be closed. This may take some time.

Aspects of common user administration


Due to the different concepts behind the Livelink and the WCM user administrations, there are restrictions and differences when compared to the usual handling of the Admin client. For administrators used to working with the Admin client, the following table provides an overview of these differences.

Table 1 – Notes on common user administration with the Admin client

User administration item	Description
Adding users	
User privileges	A user added with the Admin client has the Livelink privileges “Log-in enabled” and “Public Access enabled” and is assigned to the default group.
User name	<p>The Admin client provides only the field <i>Name</i> for the user's name. The contents of this field can be split up into the Livelink name components <i>First Name</i>, <i>Middle Initial</i>, and <i>Last Name</i>. For this purpose, first name, middle initial, and last name must be separated by at least two spaces in the Admin client.</p> <p>Example</p> <p>The entry “Joshua{two spaces}F{two spaces}Stein” in the <i>Name</i> field of the Admin client is split up as follows in Livelink:</p> <p><i>First Name:</i> Joshua <i>Middle Initial:</i> F <i>Last Name:</i> Stein</p>
Editing user information	
Finding users	You can either use the user ID or the last name of the user for searching.

User administration item	Description
Groups and roles in general	
Group and role concept	<p>Livelink provides only users and groups – no roles. For this reason, the WCM roles are represented by Livelink groups.</p> <p>For each Livelink group, the WCM User Administration module stores the information whether this group is a group or a role in the WCM system.</p>
Adding groups and roles	
Unique names	<p>In the WCM system, the names of groups and roles must only be unique within all groups or roles respectively. This means that a group can have the same name as a role. When storing the user data in Livelink, however, this is not allowed as roles are represented by groups in Livelink.</p>
Editing group and role information	
Editing WCM-specific settings	<p>In Livelink, you cannot manage additional information, such as the e-mail address, for groups. This means that you have to use the Admin client to edit WCM-specific information of groups and roles.</p>

User Administration Functions in the Admin Client

To open the user administration, click the tab  **User administration** in the Admin client. The items you can manage in this view are displayed in a tree structure.

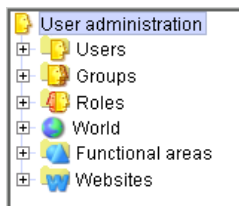


Fig. 4 – User administration items in the Admin client

The following table gives an overview of the functions available for each item.

Table 2 – Items and functions of the user administration in the Admin client

Item	Available functions
<i>User</i>	Add and configure users and edit user information Assign users to groups, roles, functional areas, and websites, as well as assign default object rights and administration rights See section “Managing Users in the Admin Client” starting on page 33
<i>Groups</i>	Add and configure groups and edit group information Assign users, functional areas, and websites to groups, and define default object rights and administration rights See section “Managing Groups and Roles in the Admin Client” starting on page 40

Item	Available functions
<i>Roles</i>	Add and configure roles and edit role information Assign users, functional areas, and websites to roles, and define default object rights and administration rights See section “Managing Groups and Roles in the Admin Client” starting on page 40
<i>World</i>	This entry represents all users of the WCM system. You can assign administration rights to it. See section “Managing the Rights of the Principal “World”” starting on page 47
<i>Functional areas</i>	Add functional areas and assign users, groups, and roles to functional areas See section “Managing Functional Areas” starting on page 47
<i>Websites</i>	Assign users, groups, and roles to websites See section “Website Assignments” starting on page 52

Finding Principals in the Admin Client

The number of principals stored in a Livelink system can be very high. Loading and displaying all principals may thus take a long time. To restrict the number of items displayed, the user administration of the Admin client provides filters for principals.

The fields for the filter criteria are displayed in the right window pane when you click one of the items *Users*, *Groups*, or *Roles*.

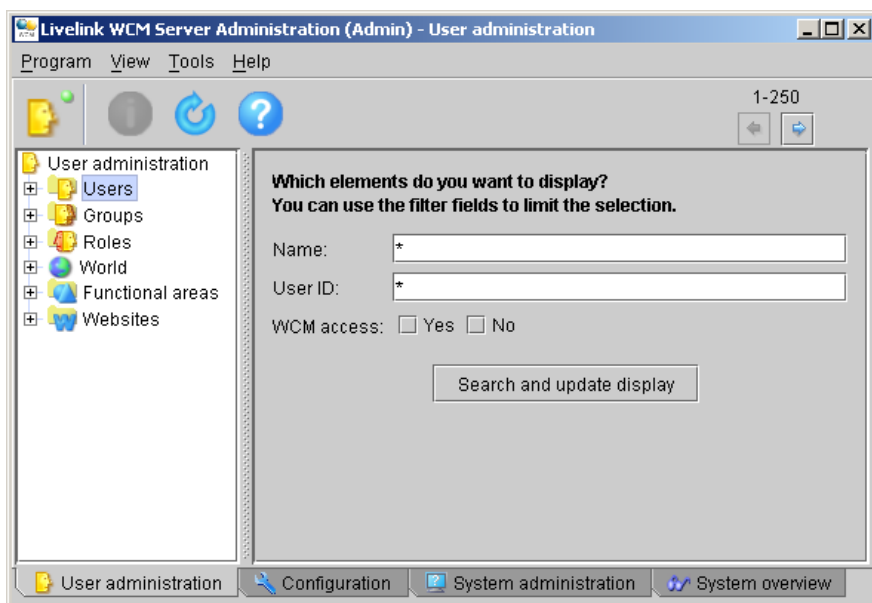


Fig. 5 – Finding principals in the Admin client

To find principals in the Admin client:

1. Enter the criteria for the display filter. You can use the following search criteria:
 - name of the principal
Only the last name can be used to search for a user.
 - user ID
 - WCM access

You can use the asterisk (*) character as a placeholder in searches. The asterisk replaces any number of characters before, after, or inside a search string. If you enter "m*" in the *User ID* field, for example, the search returns a list of all users whose IDs begin with that letter, e.g. "mmoreno" or "mrossi".

2. Confirm the search criteria with ENTER or by clicking the *Search and update display* button.

The tree on the left is updated according to the search results. For example, after a search for individual users, the *Users* item contains only the users found in the search.

3. To display all the items again, delete the filter criteria and confirm with ENTER or by clicking *Search and update display*.

Managing Users in the Admin Client

If you expand the main item *Users* in the *User administration* view, a list showing all the users stored in your Livelink system will be displayed. With the installation, all Livelink users are automatically granted access to the WCM system. However, they do not have any administration rights and functional areas assigned. Also, they only have the default object rights “Read” and “Read (Production)”. That means that the users cannot edit any website or modify the configuration of the WCM system.

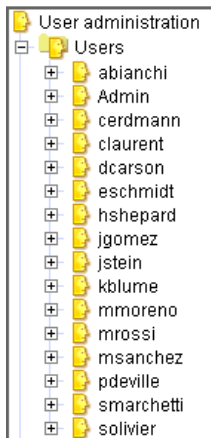


Fig. 6 – Stored users in the Admin client

The following functions are available to you for managing users in the Admin client:

- adding a user, see the following section
- editing user information (e.g. the e-mail address), see section “Editing User Information in the Admin Client” on page 36
- editing user assignments, see section “Modifying User Assignments in the Admin Client” on page 37
- deleting a user from the user administration, see section “Deleting a User in the Admin Client” on page 39

Note: Changes to a user’s settings do not take effect in the WCM system until the user logs in to the WCM system again. As an administrator, you can log out users via the System administration (for detailed information, refer to the online help of the Admin client).

Adding a User in the Admin Client

To add a user with the Admin client:

1. In the *User administration* view, select the item *Users*.
2. Choose *New user* on the context menu or click the corresponding icon.



Icon for adding a user

3. In the *New user* dialog box, make the settings for the user.

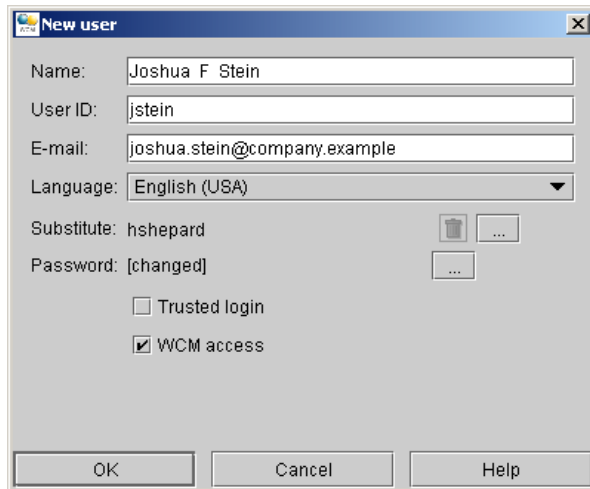


Fig. 7 – Adding a user with the Admin client

For detailed information on the individual fields, refer to the Admin client's online help.

4. Click the **OK** button.

Notes

- A user added with the Admin client has the Livelink privileges “Log-in enabled” and “Public Access enabled” and is assigned to the default group.
- The user ID is used as the Livelink log-in name.
- The Admin client provides only the field *Name* for the user's name. The contents of this field can be split up into the Livelink name components *First Name*, *Middle Initial*, and *Last Name*. For this purpose, first name, middle initial, and last name must each be separated by at least two spaces in the Admin client.

Example

The entry “Joshua{two spaces}F{two spaces}Stein” in the *Name* field of the Admin client is displayed as follows in Livelink:

First Name: Joshua

Middle Initial: F

Last Name: Stein

Editing User Information in the Admin Client

To edit the user information, e.g. the e-mail address:

1. Select the user via *User administration* → *Users* → {User ID}.

The settings for the user are displayed in the right window pane.

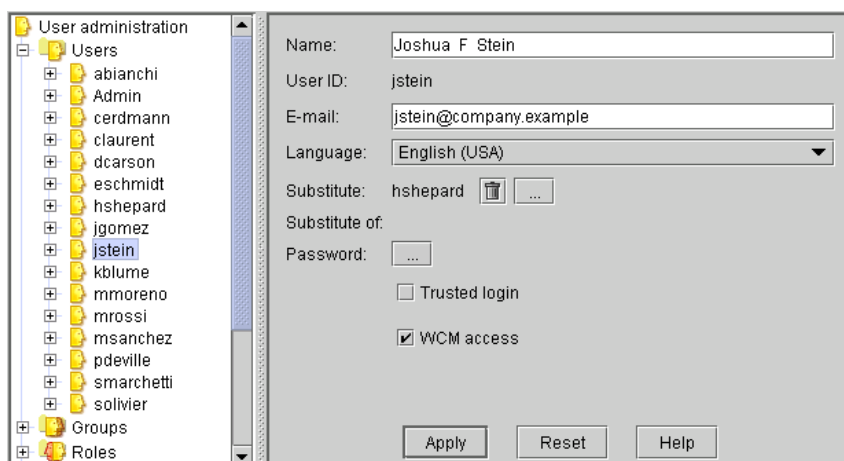


Fig. 8 – Modifying settings of a user in the Admin client

2. Make the desired changes.

For detailed information on the individual fields, refer to the Admin client's online help.

3. Click the *Apply* button.

Modifying User Assignments in the Admin Client

By means of assignments, you can assign a user to a group or role, grant the user access to a website, and set the user's default object rights and administration rights. For general information on groups, roles, functional areas, and rights, refer to the Livelink WCM Server Administrator Manual (chapter "Concepts").

To open the assignments of a user, select the respective user via *User administration* → *Users* → {user ID} and double-click the user ID. To modify the assignments, use the toolbar or the context menu.

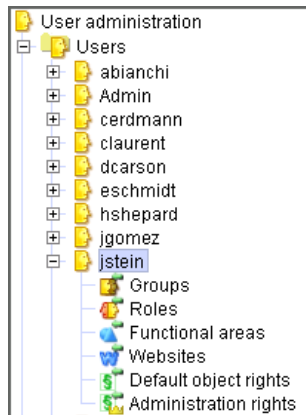


Fig. 9 – Assignments of a user in the Admin client

You can make the following assignments for a user. The overview shows the icons for creating and removing assignments between the items.



Groups: You can assign users to certain groups. Users will receive the rights of the respective group(s).



Roles: You can assign users to certain roles. Users will receive the rights of the respective role(s).



Functional areas: The functional areas are used to determine which types of WCM objects the users may add, check out, and check in, as well as which dialog boxes and functions are available in the Content client. You can assign functional areas to each user individually.



Websites: If you assign a website to a user, this user can access the objects of this website via the Content client. The exact access rights depend on the user's membership in groups or roles, the user's functional areas, and on the rights settings of the individual WCM objects.



Default object rights: For each user, you can set the default object rights for editing WCM objects. These settings are used as preset access rights when a user is added to the list of persons authorized to access an object in the Content client.

Administration rights: Here you determine the users' rights for the administration of the WCM system. The different rights can be assigned individually. In this way, you can control precisely which settings the individual users can view and what changes they can make. Depending on the assigned rights, the Admin client displays only those views and functions that the user is entitled to use. For example, if the user does not have the right to view the configuration, the Admin client hides this view.

The online help of the Admin client provides an overview of the administration rights.

Note: If a user has individual administration rights in addition to the administration rights due to the group or role membership, the rights are added up. Moreover, **all** users have the administration rights of the principal "World", see section "Managing the Rights of the Principal "World"" on page 47.

Deleting a User in the Admin Client

To delete a user in the Admin client:

1. Select the user via *User administration* → *Users* → {User ID}.
2. Choose *Delete user* on the context menu or click the corresponding icon.



Icon for deleting a user

3. Confirm the security prompt by clicking the Yes button.

The deleted user can no longer log in to the WCM system. In Livelink, the user will be marked as deleted. If you only want to prevent a user from accessing the WCM system, you can clear the check box *WCM access* in the user's settings. Thus, you do not need to delete the complete user entry.

Notes

- If you delete a user after you have added this user to the list of persons authorized to access a WCM object in the Content client, this change is not adopted automatically. You must update the list of authorized persons manually.
- As a result of deleting users, there may be WCM objects that can no longer be accessed by any principal. Choose *Collect orphan objects* on the *Tools* menu to assign such objects to a user.
- A user with administration rights can only be deleted by users that have the right "Change administration rights".

Managing Groups and Roles in the Admin Client

With Livelink WCM Server, you can assign the users of a website to groups and control access to the website objects by assigning rights to these groups. Usually, groups are tied to organizational structures, such as departments or projects, e.g. “Marketing” or “Sales”. Roles offer another way of classifying users. Unlike groups, roles are task-related, e.g. “IntranetAdmin” or “Editor”. For general information on controlling access by means of groups and roles, refer to the Livelink WCM Server Administrator Manual (chapter “Concepts”).

Notes:

Livelink provides only users and groups – no roles. For this reason, the WCM roles are represented by Livelink groups. For each Livelink group, the information is stored whether this group is a group or a role in the WCM system.

In Livelink, you cannot manage additional information, such as the e-mail address, for groups. This means that you have to use the Admin client to edit WCM-specific information of groups and roles.

The procedures for managing groups and roles in the Admin client are identical and are therefore described together.

If you open the main item *Groups* or *Roles* in the *User administration* view, the available groups or roles are shown in the tree on the left.

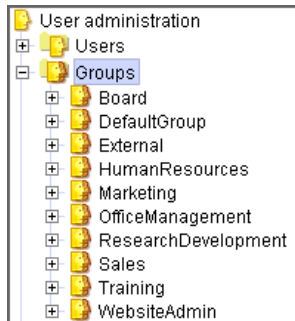


Fig. 10 – Stored groups in the Admin client

The following functions are available to you for managing groups and roles:

- adding a group or role, see the following section
- changing the e-mail address or the WCM access of groups or roles, see “Editing Group or Role Information in the Admin Client” on page 43
- editing the assignments of a group or role, see section “Modifying the Assignments of a Group or Role in the Admin Client” on page 43
- removing a group or role from the user administration, see section “Deleting a Group or Role in the Admin Client” on page 46

Note: Changes to a group or role do not take effect in the WCM system until the corresponding users log in to the WCM system again. As an administrator, you can log out users via the System administration (for detailed information, refer to the online help of the Admin client).

Adding a Group or Role in the Admin Client

To add a group or role in the Admin client:

1. In the *User administration* view, select the item *Groups* or *Roles*.
2. Choose *New group* or *New role* on the context menu or click the corresponding icon.



Icon for adding a group



Icon for adding a role

3. In the following dialog box, make the settings for the group or role.

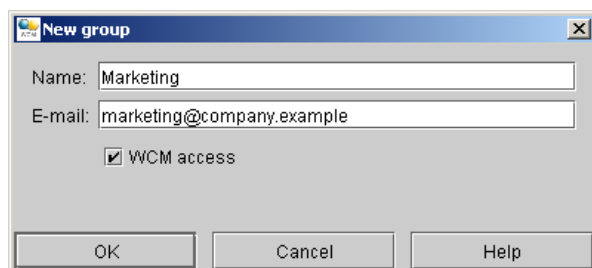


Fig. 11 – Adding a group in the Admin client

For detailed information on the individual fields, refer to the Admin client's online help.

Note: In the WCM system, the names of groups and roles must only be unique within all groups or roles respectively. This means that a group can have the same name as a role. If the user information is stored in Livelink, this is **not** allowed, as roles are treated like groups in Livelink.

4. Confirm the entries by clicking the *OK* button.

Editing Group or Role Information in the Admin Client

To change the settings of a group or role:

1. Select the desired item via *User administration* → *Groups or Roles* → *{group name/role name}* .

The settings are displayed in the right window pane.

2. Make the desired changes.

For detailed information on the individual fields, refer to the Admin client's online help.

3. Click the *Apply* button.

Note: In Livelink, you cannot manage additional information, such as the e-mail address, for groups. This means that you have to use the Admin client to edit WCM-specific information of groups and roles.

Modifying the Assignments of a Group or Role in the Admin Client

By means of the assignments, you can assign users to a group or role, allow the group/role access to a website, and set the default object rights and administration rights for the group/role. For general information on groups, roles, functional areas, and rights, refer to the Livelink WCM Server Administrator Manual (chapter “Concepts”).

To open the assignments of a group or role, select *User administration* → *Groups or Roles* → *{group name/role name}*. Use the toolbar or the context menu to modify the assignments.

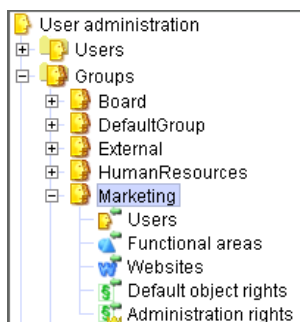


Fig. 12 – Assignments of a group in the Admin client

You can make the following assignments for a group or role. The overview shows the icons for creating and removing assignments between the items.



Users: From the list of stored users, you can select which users are to belong to a certain group or role.



Functional areas: The functional areas are used to determine which types of objects the users may add, check out, and check in, as well as which dialog boxes and functions are available in the Content client. If you assign functional areas to a group or role, all users belonging to this group or role get the respective rights.



Websites: If you assign a group or role to a website, the respective users can access the objects of this website via the Content client. The exact access rights depend on the group and role settings, the assigned functional areas, and the rights settings of the individual WCM objects.



Default object rights: For each group or role, you can set the default object rights for editing WCM objects. These settings are used as preset access rights when a group or role is added to the list of persons authorized to access an object in the Content client.

Administration rights: Here you determine the rights of the group or role for the administration of the WCM system. The different rights can be assigned individually. In this way, you can control precisely which settings the assigned users can view and what changes they can make. In accordance with the rights assigned, the Admin client only displays those views and functions that the assigned users are entitled to use. For example, if the group or role does not have the right to view the configuration, this view is hidden in the Admin client.

The online help of the Admin client provides an overview of the administration rights.

Note: If a user has individual administration rights in addition to the administration rights due to the group or role membership, the rights are added up. Moreover, **all** users have the administration rights of the principal “World”, see section “Managing the Rights of the Principal “World”” on page 47.

Deleting a Group or Role in the Admin Client

To delete a group or role in the Admin client:

1. Select the group or role via *User administration* → *Groups* or *Roles* → {group name/role name}.
2. Choose *Delete group* or *Delete role* on the context menu or click the corresponding icon.



Icon for deleting a group



Icon for deleting a role

3. Confirm the security prompt by clicking the *Yes* button.

Notes

- If you delete a group or role after you have added it to the list of persons authorized to access a WCM object in the Content client, this change is not adopted automatically. You must update the list of authorized persons manually.
- As the result of deleting groups or roles, there may be WCM objects that can no longer be accessed by any principal. Choose *Collect orphan objects* on the *Tools* menu to assign such objects to a user.
- A group or role with administration rights can only be deleted by users that have the right “Change administration rights”.

Managing the Rights of the Principal “World”

The principal “World” represents all users of the WCM system. Technically speaking, it is a group. The administration rights of this principal are valid for all users of the WCM system in addition to the administration rights that have been set for the individual users or for the assigned group or role.

Note: By default, all users (i.e. “World”) have only access to the user administration in the Admin client. We recommend that you do not change this setting in order to reserve access to the configuration and system administration to administrators.

When you modify the administration rights of the principal “World”, you proceed in the same way as for users and groups.

The online help of the Admin client provides an overview of the administration rights.

Managing Functional Areas

Functional areas perform two essential tasks:

- They determine which types of objects the users may add, check out, and check in. Some functional areas, such as “Basic”, are by default assigned to object types. Only users that have the corresponding functional area can add, check in, and check out objects of this object type.
- They determine which views and dialog boxes are available to the user in the Content client.

For general information on functional areas, refer to the Livelink WCM Server Administrator Manual (chapter “Concepts”).

If you open the main item *Functional areas* in the *User administration* view, a list of available functional areas is displayed.

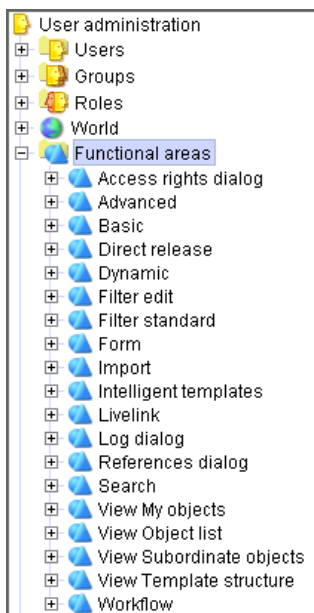


Fig. 13 – Available functional areas

The following functions are available to you:

- adding functional areas, see the following section
- viewing the settings of functional areas, see “Settings of Functional Areas” on page 50
- assigning functional areas to users, groups, and roles, see “Assigning Functional Areas” on page 51
- deleting functional areas, see “Deleting a Functional Area” on page 52

Functional areas can also be assigned to object types. This is done in the settings of the object types. By this assignment, you determine that objects with this object type may only be added, checked out, and checked in by users with a certain functional area.

Adding a Functional Area

To add a functional area:

1. In the *User administration* view, select the item *Functional areas*.
2. Choose *New functional area* on the context menu or click the corresponding icon.



Icon for adding a functional area

3. In the following dialog box, enter the name for the new functional area.

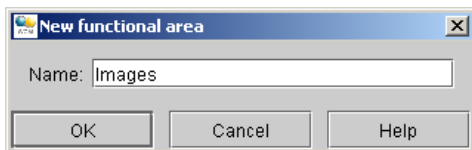


Fig. 14 – Adding a functional area

4. Confirm the settings by clicking the *OK* button.

Notes:

If you want the name of the functional areas to be displayed in the language of the users, you can translate it. For more information, refer to the Livelink WCM Server Administrator Manual (chapter “Working in the Admin Client”).

A new functional area must be assigned to a principal in order to take effect. Moreover, it must be assigned to one or more object types.

Settings of Functional Areas

Note: For displaying the settings of the functional areas, you require the administration right “Access to configuration”.

To view the settings of a functional area, select it via *User administration* → *Functional areas* → {name of the functional area}. In the right window pane, the name, the website, and the assigned object types are displayed.

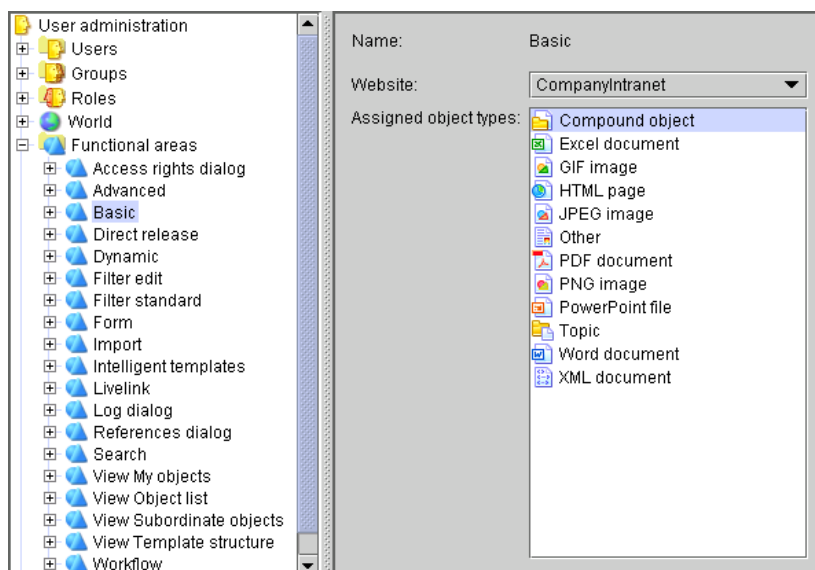


Fig. 15 – Settings of functional areas

In the *Configuration* view, you can change the assignments between object types and functional areas and make new assignments for custom functional areas and object types.

Assigning Functional Areas

If you open one of the functional areas, the list of assignments opens in the tree: *Users*, *Groups*, and *Roles*. If you open one of the assignment items, e.g. *Users*, the right window pane shows the list of assigned items, e.g. the users who are assigned to the functional area “Direct release”.

By means of the toolbar or the context menu, you can make and remove the following assignments:



Assign users and remove assignments



Assign groups and remove assignments



Assign roles and remove assignments

If you assign a user, group, or role to a functional area, the respective users get the rights associated with this functional area (e.g. the right for direct release or the right to add, check out, and check in JSP pages).

If a user is assigned to one functional area due to group or role assignments and to another functional area due to individual assignments, both functional areas apply to the user. That means that the rights are “added up”.

Note: If you modify the functional area assignments, the changes do not take effect until the relevant users log in to the WCM system again. As an administrator, you can log out users via the System administration (for detailed information, refer to the online help of the Admin client).

Deleting a Functional Area

Note: You can only delete custom functional areas that no longer have any users, groups, or roles assigned. Moreover, this functional area must not be assigned to an object type.

To delete a functional area:

1. Select the functional area via *User administration* → *Functional areas* → {name of functional area}.
2. Choose *Delete functional area* on the context menu or click the corresponding icon.



Icon for deleting a functional area

Website Assignments

In the *User administration* view, you can use the main item *Websites* to assign users, groups, and roles to websites. To create and configure websites, change to the *Configuration* view of the Admin client. For more information, refer to the Livelink WCM Server Administrator Manual (chapter “Managing Websites”).

Open the main item *Websites* to display the list of assignments to users, groups, and roles.

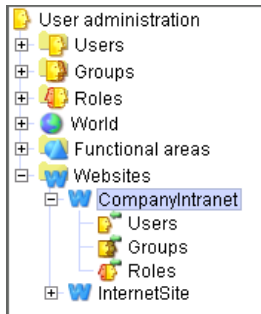


Fig. 16 – Website assignments

By means of the toolbar or the context menu, you can make and remove the following assignments:



Assign users and remove assignments



Assign groups and remove assignments



Assign roles and remove assignments

If you assign a website to a user, group, or role, the respective users can access the objects of this website via the Content client. The exact access rights depend on the users' memberships in groups and roles, the users' functional areas, and on the rights settings of the individual WCM objects.

Note: If you modify the website assignments, the changes do not take effect until the relevant users log in to the WCM system again. As an administrator, you can log out users via the System administration (for detailed information, refer to the online help of the Admin client).

Managing Principals in Livelink

Note: A user with the appropriate Livelink privileges can make changes to the user information which also affect the WCM system. These users do not require any administration rights in the WCM system for this.

You can manage all users who are to access the WCM system and the Livelink system, in the Livelink user administration. To group users according to the company's organizational structure and to assign appropriate privileges, you can create groups. For detailed information on privileges, refer to the Livelink online help.

Opening the Livelink user administration

To open the Livelink user administration:

1. Enter the URL of the Livelink system in the address field of your browser.

The URL is of type

<http://livelinkserver.company.example/Livelink/livelink.exe>



Fig. 17 – Logging in to the Livelink system

2. Log in to the Livelink system as an administrator.

Note: Both user ID and password are case sensitive. Correct use of uppercase and lowercase letters is essential. For Livelink, “Admin” and “admin” are two different users.

3. Choose *Users & Groups* on the *Enterprise* menu.


The Users and Groups page opens.

Aspects of common user administration

Due to the different concepts behind the Livelink and the WCM user administrations, there are restrictions and differences when compared to the usual handling of the Livelink user administration. For administrators used to working with the Livelink user administration, the following table provides an overview of these differences.

Table 3 – Notes on managing users with Livelink

User administration item	Description
Adding users	
User privileges	Provided that the user has the Livelink privilege “Log-in enabled”, the user can also log in to the WCM system as the <i>WCM Access</i> check box is selected by default. In the WCM system, the user initially has the access rights “Read” and “Read (Production)”.

User administration item	Description
User name and log-in name	A WCM user must have a name in addition to the user ID. For users who only have a log-in name in Livelink, the log-in name is used as both the user ID and the name in the Admin client.
	Livelink provides three fields for the name of a user: <i>First Name</i> , <i>Middle Initial</i> , and <i>Last Name</i> (e.g. "Joshua", "F", and "Stein"). The contents of the <i>Name</i> field in the Admin client is built from these name components ("Joshua F Stein").
E-mail address	<p>A WCM user needs a valid e-mail address. If you want to use the notification functions of the WCM system, you should specify an e-mail address for all users in Livelink.</p> <p>The Admin client assigns a default e-mail address to all users who are stored in the Livelink database without an e-mail address. For this purpose, the system uses the e-mail address of the person who is to be notified if the WCM licenses expire or are exceeded. This e-mail address is not stored in the Livelink data storage.</p> <p>To use a different default e-mail address, you have to edit the KUAF table in the Livelink server's database. Use the following SQL command:</p> <pre>update KUAF set MAILADDRESS = '{default e-mail address}' where TYPE = 0 and MAILADDRESS is null</pre>
Editing user information	
Editing WCM-specific settings	Use the  tab to edit the WCM-specific user settings in Livelink.

User administration item	Description
Managing groups	
Handling of roles	<p>The WCM system differs between groups and roles, whereas Livelink only provides groups. For this reason, the WCM roles are represented by Livelink groups. Thus, you cannot distinguish between WCM groups and roles in Livelink.</p> <p>Use the Admin client for editing roles.</p>

Finding Principals in Livelink

The search functions of the Livelink user administration enable you to quickly find the right entry, even with a large number of users and groups.

To find principals in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. On the Users and Groups page, click the desired search parameter in the *Find* drop-down list.

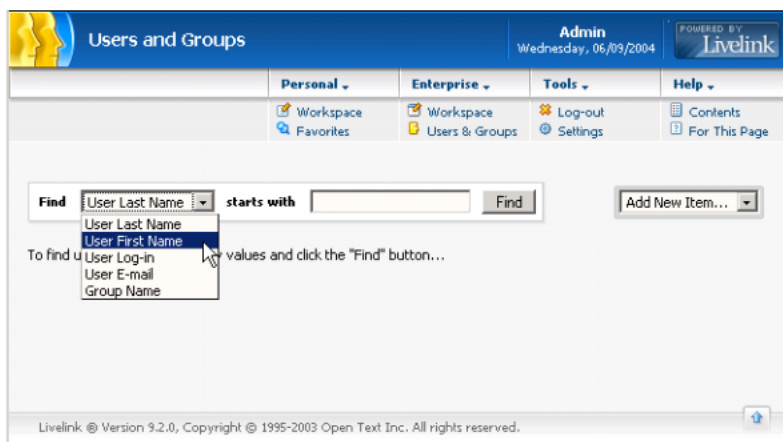


Fig. 18 – Finding principals in Livelink

You can use the following search parameters:

- *User Last Name* (default setting)
- *User First Name*
- *User Log-in*
- *User E-mail*
- *Group Name*

3. Type a search term in the *starts with* field.

Notes:

The value you supply for the search parameter is **not** case sensitive. For example: If you enter “ma” for the first name of a user, information on all users whose first names start with “ma” or “Ma” is displayed.

If you leave the *starts with* field blank, Livelink retrieves all users or groups in the system.

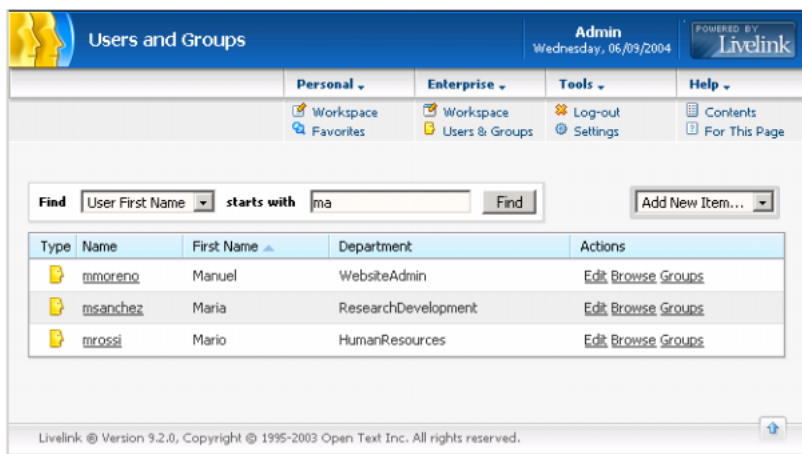
4. Click the *Find* button.

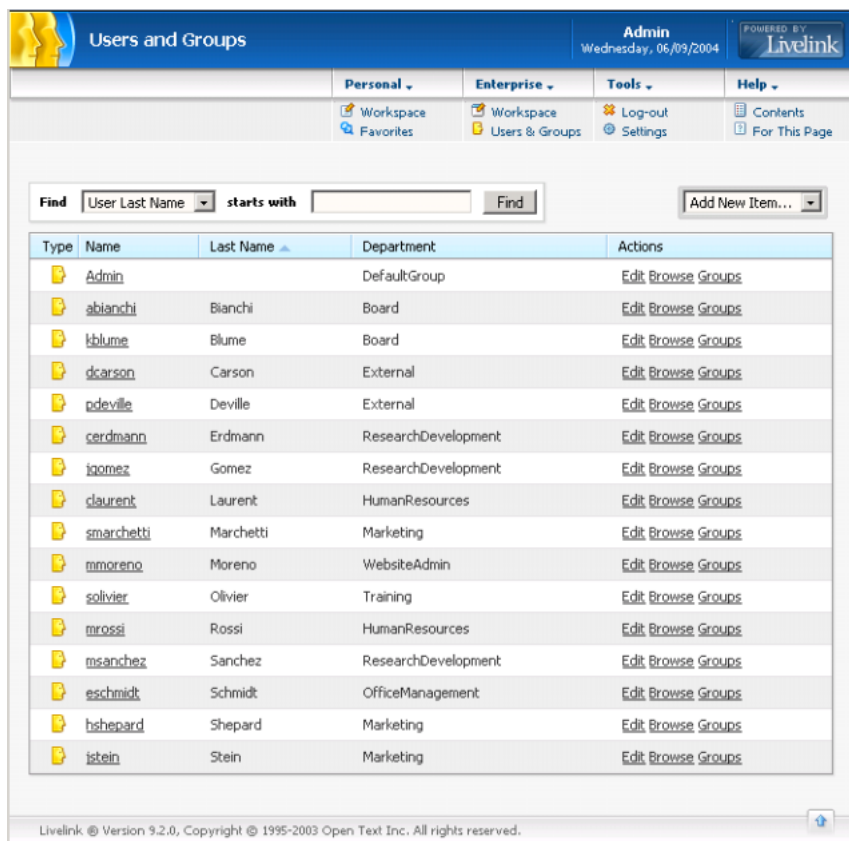
Fig. 19 – Users found in Livelink

The search results are displayed on the Users and Groups page. Provided that you have the appropriate privileges, the *Actions* column contains the following links.

- **Edit:** Click this link to modify the settings of a principal.
- **Browse:** Click this link to view the Personal Workspace of a user.
- **Groups:** Click this link to view information on the group(s) that the user belongs to.

Managing Users in Livelink

If you leave the field *starts with* on the Users and Groups page blank and click the *Find* button, Livelink retrieves all users in the system.



The screenshot shows the 'Users and Groups' interface in Livelink. The top navigation bar includes 'Personal', 'Enterprise', 'Tools', and 'Help' menus. Below these are links for 'Workspace', 'Favorites', 'Log-out', 'Settings', 'Contents', and 'For This Page'. The main section features a search bar with a 'Find' button and a dropdown menu for 'User Last Name'. A table below the search bar lists users with columns for 'Type', 'Name', 'Last Name', 'Department', and 'Actions'. The table contains 16 rows of user data. At the bottom, there is a footer with the text 'Livelink © Version 9.2.0, Copyright © 1995-2003 Open Text Inc. All rights reserved.'

Type	Name	Last Name	Department	Actions
	Admin		DefaultGroup	Edit Browse Groups
	abianchi	Bianchi	Board	Edit Browse Groups
	kblume	Blume	Board	Edit Browse Groups
	dcarson	Carson	External	Edit Browse Groups
	pdeville	Deville	External	Edit Browse Groups
	cerdmann	Erdmann	ResearchDevelopment	Edit Browse Groups
	lgomez	Gomez	ResearchDevelopment	Edit Browse Groups
	claurent	Laurent	HumanResources	Edit Browse Groups
	smarchetti	Marchetti	Marketing	Edit Browse Groups
	mmoreno	Moreno	WebsiteAdmin	Edit Browse Groups
	solvier	Olivier	Training	Edit Browse Groups
	mrossi	Rossi	HumanResources	Edit Browse Groups
	msanchez	Sanchez	ResearchDevelopment	Edit Browse Groups
	eschmidt	Schmidt	OfficeManagement	Edit Browse Groups
	hshepard	Shepard	Marketing	Edit Browse Groups
	jstein	Stein	Marketing	Edit Browse Groups

Fig. 20 – Users stored in Livelink

The following functions are available to you for managing users:

- adding a user, see the following section
- edit user information (see section “Editing User Information in Livelink” on page 64)
- deleting a user from the user administration, see section “Deleting a User in Livelink” on page 71

Adding a User in Livelink

To add a user in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. On the *Users & Groups* page, choose *User* on the *Add New Item* menu.
3. Define the settings for the user on the *Add New User* page.

Log-in Name: ⚠	<input type="text" value="jstein"/>
Department: ⚠	<input type="text" value="Training"/>
Password:	<input type="password" value="*****"/>
Verify Password:	<input type="password" value="*****"/>
First Name:	<input type="text" value="Joshua"/>
Middle Initial:	<input type="text" value="F"/>
Last Name:	<input type="text" value="Stein"/>
Title:	<input type="text"/>
E-mail:	<input type="text" value="joshua.stein@company.example"/>
Phone:	<input type="text"/>
Fax:	<input type="text"/>
Office Location:	<input type="text"/>
Time Zone:	<input type="text"/>
Privileges:	<div><input checked="" type="checkbox"/> Log-in enabled <input checked="" type="checkbox"/> Public Access enabled <input type="checkbox"/> Can create/modify users <input type="checkbox"/> Can create/modify groups <input type="checkbox"/> User administration rights <input type="checkbox"/> System administration rights</div>
<div><input type="button" value="Submit"/> <input type="button" value="Reset"/></div>	

Fig. 21 – Adding a user in Livelink

For detailed information on the individual fields, refer to the Livelink online help.

4. Click the *Submit* button.

Notes

- Provided that the user added to Livelink has the privilege “Log-in enabled”, the user can also log in to the WCM system as the WCM access is enabled by default. In the WCM system, the user initially has the access rights “Read” and “Read (Production)”.
- The log-in name is used as the user ID in the WCM system. A WCM user must have a name in addition to the user ID. For users who only have a log-in name in Livelink, the log-in name is used as **both** the user ID **and** the name in the Admin client.
- Livelink provides three fields for specifying the name of a user: *First Name*, *Middle Initial*, and *Last Name* (e.g. “Joshua”, “F”, and “Stein”). The contents of the *Name* field in the Admin client is built from these name components (“Joshua F Stein”).
- In order to change the WCM administration rights of users, you have to use the Admin client.
- A WCM user needs a valid e-mail address. If you want to use the notification functions of the WCM system, you should specify an e-mail address for each user in Livelink.

The Admin client assigns a default e-mail address to all users who are stored in the Livelink database without an e-mail address. For this purpose, the system uses the e-mail address of the person who is to be notified if the WCM licenses expire or are exceeded. This e-mail address is not stored in the Livelink data storage.

To use a different default e-mail address, you have to edit the KUAF table in the Livelink server's database. Use the following SQL command:

```
update KUAF set MAILADDRESS = '{default e-mail address}' where  
TYPE = 0 and MAILADDRESS is null
```

Editing User Information in Livelink

In Livelink, all information about a user is stored in the so-called user profile. To view the user profile, click the user's log-in name on the Users and Groups page.

User information is stored on the following tabs:



The *General* tab contains the settings which were made when the user was created.



The *Personal* tab contains items as a photograph, birthday, and personal interests. All entries on this tab are optional.



The *WCM* tab is only available if you use a common user administration for Livelink and the WCM system. This tab contains the user's WCM-specific settings, such as access rights, as well as assignments to functional areas and websites.

The Admin user is allowed to edit all user profiles. Users who have the "User administration rights" privilege are allowed to edit all profiles with the exception of the Admin user. Users with the privilege "Can create/modify users" are allowed to change the profiles of the users they create.

To change the user information on one of the tabs in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. Use the search functions on the Users and Groups page to find the user (see section "Finding Principals in Livelink" on page 57).
3. Click the user's *Edit* link in the *Actions* column.
4. Select one of the tabs, e.g. the *WCM* tab.

General Personal WCM	
WCM access:	<input checked="" type="checkbox"/>
Trusted login:	<input type="checkbox"/>
Language:	English (USA) ▼
Substitute:	<input type="text"/> ... Remove
Websites:	<input checked="" type="checkbox"/> CompanyIntranet <input checked="" type="checkbox"/> InternetSite
Functional areas:	<input checked="" type="checkbox"/> Access rights dialog <input checked="" type="checkbox"/> Advanced <input checked="" type="checkbox"/> Basic <input type="checkbox"/> Direct release <input checked="" type="checkbox"/> Dynamic <input type="checkbox"/> Filter edit <input checked="" type="checkbox"/> Filter standard <input type="checkbox"/> Form <input checked="" type="checkbox"/> Import <input type="checkbox"/> Intelligent templates <input checked="" type="checkbox"/> Livelink <input type="checkbox"/> Log dialog <input checked="" type="checkbox"/> References dialog <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> View My objects <input type="checkbox"/> View Object list <input type="checkbox"/> View Subordinate objects <input type="checkbox"/> View Template structure <input checked="" type="checkbox"/> Workflow
Default object rights:	Create Yes ▼ Read Yes ▼ Change object Yes ▼ Delete Yes ▼ Move and copy Yes ▼ Release No ▼ Change metadata Yes ▼ Change rights No ▼ Read (production) Yes ▼
Apply Reset Cancel	


Fig. 22 – Editing user information in Livelink

5. Make the desired changes.

For information on the individual fields on the *General* and *Personal* tabs, refer to the Livelink online help. The settings on the *WCM* tab are described in the following section.

6. Click the *Apply* button.

Settings on the *WCM* tab

- *WCM access*: If you want to block the user's access to the WCM system, clear this check box. In this way, you prevent a user from accessing the WCM system without having to delete the user profile. The user can still log in to Livelink.
- *Trusted login*: Here you can define whether this user is allowed to perform a trusted login. If you select this check box, the user does not need any further authentication for accessing other WCM products after initial login to the WCM system.
- *Language*: the preferred language of the user. Click the desired entry in the *Language* drop-down list. In the Configuration dialog box of the Content client, the user can also select a language for the client. This setting takes priority over the setting on this tab and cannot be changed by the administrator.
- *Substitute*: Click the  button to select a substitute for this user from a list. In the Content client, the user selected here can log in as the substitute for the user. The substitute may perform all functions for which the user is authorized.

Click the *Remove* button to unassign the substitute.

- *Websites*: If you want to change the assigned website(s) of the user, select or clear the check boxes of the website(s).

The assignment to a website enables the user to access the website objects with the Content client.

- **Functional areas:** The functional areas are used to determine which types of WCM objects the user may edit and which dialog boxes and functions are available in the Content client.

If you want to change the assigned functional area(s) of the user, select or clear the check boxes of the functional area(s). The following table shows the default functional areas available after the installation of the WCM system.

Table 4 – Available functional areas

Functional area	Default assignment to object type	View or function in the Content client
<i>Basic</i>	Compound object, Excel document, GIF image, HTML page, JPEG image, Other, PDF document, PNG image, PowerPoint file, Topic, Word document, XML document	Add, check out, and check in objects based on these object types
<i>Advanced</i>	Frame, Frame topic, HTML template, XML template	
<i>Dynamic</i>	Assembled object, ASP page, ASP template, ASP topic, CGI script, Form template, JavaScript, JSP page, JSP template, JSP topic, PHP page, PHP template, PHP topic, XSLT document, XSLT template	Add, check out, and check in objects based on these object types
<i>Form</i>	Form instance	

Functional area	Default assignment to object type	View or function in the Content client
<i>Workflow</i>	Workflow	Assign workflows to objects, remove workflow assignments
<i>Intelligent templates</i>	none	Not used by default, required for compatibility with VIP 5e
<i>Direct release</i>		Work with the <i>Direct release</i> option.
<i>References dialog</i>		View the References dialog box
<i>Access rights dialog</i>		View the Access rights dialog box

Functional area	Default assignment to object type	View or function in the Content client
<i>Log dialog</i>	none	View the Log dialog box
<i>Filter standard</i>		Use the standard filters
<i>Filter edit</i>		Create and edit filters in the filter editor Note: Additionally, the functional area “Filter standard” is required.
<i>View Subordinate objects</i>		View the subordinate objects of a WCM object
<i>View Object list</i>		“Object list” view
<i>View My objects</i>		“My objects” and “My work list” views
<i>View Template structure</i>		“Template structure” view
<i>Import</i>		Use import functions
<i>Search</i>		Use search functions
<i>Livelink</i>		View the Livelink metadata dialog box

- **Default object rights:** These settings are used as preset access rights when a user is added to the list of persons authorized to access an WCM object in the Content client.

If you want to change one or several default object rights, click *Yes* (object right is explicitly allowed), *No* (object right is explicitly forbidden), or the empty entry (object right not set) in the drop-down list of the right.

The following table gives an overview of the actions you can perform on WCM objects and the required access rights. For detailed information on object rights, refer to the Content Client User Manual (chapter “Concepts”).

Table 5 – Actions and required default object rights

Action	Required object rights
Add object	Read + Create + Change object (for the parent topic)
Delete object	Read + Delete + Change object
Submit object	Read + Change object
Edit an object (check out, change, check in)	Read + Change object
Change an object's metadata	Read + Change metadata
Change an object's references	Read + Change metadata
Rename object	Read + Change metadata
Move object	Read + Move and copy + Change metadata (for the source object)
Copy object	Read + Move and copy + Change metadata (for the source object)
Insert object (after cutting or copying)	Read + Create (for the target topic)

Action	Required object rights
Check access rights of object	Read
Change access rights of object	Read + Change rights + Change metadata
Release or reject object	Read + Release
Release object directly	Read + Change object
Destroy object	Read + Release

Additional information on the *WCM* tab

- The lists of websites and functional areas are updated every time the WCM Administration server is started and the configuration is changed. For this reason, the *WCM* tab does not display any websites and functional areas directly after installing the WCM User Administration module. The list will not be updated before you install the WCM system and create websites.
- You can add functional areas with the Admin client. Livelink initially shows the internal names of these custom functional areas. To display the names of the functional areas in the user's language, create translations with the Livelink Builder.

Deleting a User in Livelink

The Admin user and all users with the privilege “User administration rights” are allowed to delete any user. Also, users with the privilege “Can create/modify users” are allowed to delete the users they create.

To delete a user in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. Use the search functions on the Users and Groups page to find the user (see section “Finding Principals in Livelink” on page 57).

3. Click the user's *Edit* link in the *Actions* column.
4. Click the *Delete User* button.
5. Confirm the security prompt by clicking the *OK* button.

The user is removed from all groups of which the user was a member. The user's profile is not removed from the database, but is marked as deleted so that the user's information will still be displayed when viewing audit logs and item permissions.

Managing Groups in Livelink

Notes:

The WCM system differs between groups and roles, whereas Livelink only provides groups. For this reason, the WCM roles are represented by Livelink groups. Thus, you cannot distinguish between WCM groups and roles in Livelink.

Use the Admin client for editing roles.

To open a list of the groups that exist in the Livelink system: Choose *Users & Groups* on the *Enterprise* menu. On the Users and Groups page, click *Group Name* in the *Find* drop-down list and click the *Find* button. Directly after the installation of Livelink, only the “DefaultGroup” exists. By default, this is the department group of the Admin user.











Find	Group Name ▾	starts with	<input type="text"/>	Find	Add New Item... ▾
Type	Name ▲	Actions			
	Board	Edit			
	DefaultGroup	Edit			
	External	Edit			
	HumanResources	Edit			
	Marketing	Edit			
	OfficeManagement	Edit			
	ResearchDevelopment	Edit			
	Sales	Edit			
	Training	Edit			
	WebsiteAdmin	Edit			

Fig. 23 – Groups stored in Livelink

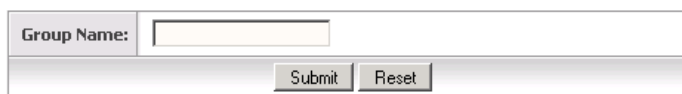
The following functions are available to you for managing groups:

- adding a group, see the following section
- editing group information (see section “Editing Group Information in Livelink” on page 74)
- removing a group from the user administration, see section “Deleting a Group in Livelink” on page 79

Adding a Group in Livelink

To add a group in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. On the *Users & Groups* page, choose *Group* on the *Add New Item* menu.
3. Type a unique name for the new group in the *Group Name* field.



A web form for adding a group. It features a label "Group Name:" followed by a text input field. Below the input field are two buttons: "Submit" and "Reset".

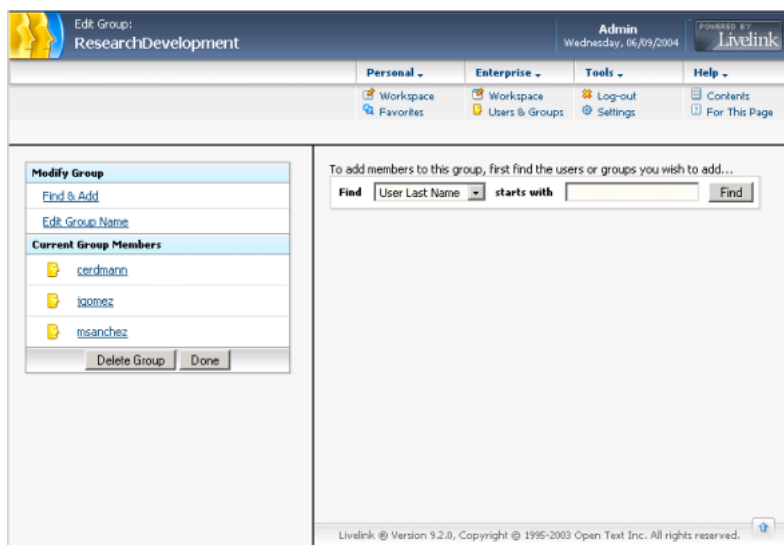
Fig. 24 – Adding a group in Livelink

4. Click the *Submit* button.

Editing Group Information in Livelink

The Edit Group page allows you to change the settings of a group. To open this page:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. Use the search functions on the Users and Groups page to find the group (see section “Finding Principals in Livelink” on page 57).
3. Click the group's *Edit* link in the *Actions* column.



A screenshot of the Livelink 'Edit Group' page. The page title is 'Edit Group: ResearchDevelopment'. The top navigation bar includes 'Admin' (Wednesday, 06/09/2004) and 'Powered by Livelink'. Below the navigation bar are four tabs: 'Personal', 'Enterprise', 'Tools', and 'Help'. The 'Enterprise' tab is selected, showing 'Workspace', 'Users & Groups', 'Log-out', and 'Settings'. The 'Tools' tab shows 'Log-out' and 'Settings'. The 'Help' tab shows 'Contents' and 'For This Page'. The main content area is divided into two sections. The left section, titled 'Modify Group', contains links for 'Find & Add', 'Edit Group Name', and 'Current Group Members'. Below these links is a list of group members: 'cerdmann', 'lgomez', and 'msanchez'. At the bottom of this section are 'Delete Group' and 'Done' buttons. The right section, titled 'To add members to this group, first find the users or groups you wish to add...', contains a search form with a 'Find' button, a dropdown menu for 'User Last Name', a 'starts with' label, and a text input field. The footer of the page reads 'Livelink © Version 9.2.0, Copyright © 1995-2003 Open Text Inc. All rights reserved.'

Fig. 25 – Editing group information in Livelink

You can do the following on the Edit Group page:

- add a user or subgroup to a group
- set the group leader
- change a group name
- delete a member from a group

Adding a member to a group in Livelink

To add a new member (user or group) to a group in Livelink:

1. On the Edit Group page, click the *Find & Add* link to display the Search bar in the right frame, if not present.
2. Use the Search bar to display the users or groups you want to add as new members to the group.
3. Select the user's or group's *Add to group* check box.

















Type	Name	Last Name	Department	Actions
	Admin		DefaultGroup	<input type="checkbox"/> Add to group
	abianchi	Bianchi	Board	<input checked="" type="checkbox"/> Add to group
	kblume	Blume	Board	<input type="checkbox"/> Add to group
	dcarson	Carson	External	<input type="checkbox"/> Add to group
	pdeville	Deville	External	<input type="checkbox"/> Add to group
	cerdmann	Erdmann	ResearchDevelopment	
	igomez	Gomez	ResearchDevelopment	
	claurent	Laurent	HumanResources	<input type="checkbox"/> Add to group
	smarchetti	Marchetti	Marketing	<input type="checkbox"/> Add to group
	mmoreno	Moreno	WebsiteAdmin	<input type="checkbox"/> Add to group
	solivier	Olivier	Training	<input checked="" type="checkbox"/> Add to group
	mrossi	Rossi	HumanResources	<input type="checkbox"/> Add to group
	msanchez	Sanchez	ResearchDevelopment	
	eschmidt	Schmidt	OfficeManagement	<input type="checkbox"/> Add to group
	hshepard	Shepard	Marketing	<input type="checkbox"/> Add to group
	istein	Stein	Marketing	<input checked="" type="checkbox"/> Add to group
				<input type="button" value="Submit"/>

Fig. 26 – Adding members to a group in Livelink

4. Click the *Submit* button.

On the left side of the window, the names of the new group members are displayed.

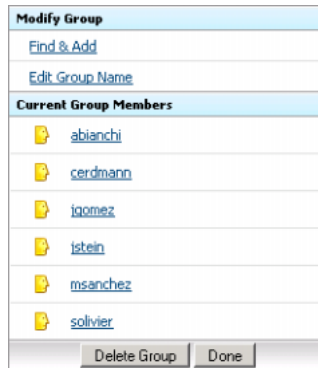


Fig. 27 – Group members in Livelink

Setting the group leader in Livelink

Any group member can be selected as the group leader. Every group can have exactly one group leader. Setting a new group leader automatically unsets the previous group leader. The previous group leader returns to regular member status.

The group leader always has the privilege to edit or delete the group. The leader does not require the privilege “Can create/modify groups” or “User administration rights” for this purpose.

To set the group leader in Livelink:

1. In the *Current Group Members* section of the Edit Group page, click the name link of the member that you want to set as the group leader.

The user information is displayed in the right window pane.



 abianchi	
Log-in Name:	abianchi
Name:	abianchi
Department:	Board
E-Mail:	abianchi@company.example
<div>Set as Group Leader Remove From Group</div>	

Fig. 28 – Setting a user as the group leader in Livelink


2. Click the *Set as Group Leader* button.

A Leader icon  replaces the User icon  to identify the user as the group leader.

Unsetting the group leader in Livelink

To unset the group leader in Livelink:

1. In the *Current Group Members* section of the Edit Group page, click the name link of the group leader.
2. Click the *Unset as Group Leader* button in the right window pane.

The user returns to regular member status and is marked with the User icon .

Changing a group name in Livelink

Important! Changing a group name in Livelink corresponds in the WCM system to deleting the group and adding a new group with identical settings. The WCM objects' access control lists which contained the original group are not modified automatically.

To change a group name in Livelink:

1. On the Edit Group page, click the *Edit Group Name* link.
2. Type the new group name in the *Group Name* field.

3. Click the *Submit* button.

Removing a Member from a Group in Livelink

Note: You cannot remove a member from his or her base group.

To remove a member from a group in Livelink:

1. In the *Current Group Members* section of the Edit Group page, click the name link of the user that you want to remove from the group.
2. Click the *Remove From Group* button.

Deleting a Group in Livelink

To delete a group in Livelink:

1. Choose *Users & Groups* on the *Enterprise* menu.
2. Use the search functions on the Users and Groups page to find the group.
3. Click the group's *Edit* link in the *Actions* column.
4. Click the *Delete Group* button.
5. Confirm the security prompt by clicking the *OK* button.

The deleted group is removed from all groups of which the group was a member. The members are removed from the deleted group. The users themselves are not deleted, however.

If you delete a group which is the department group for some users, these users automatically become members of the default group. For this reason, you should **never** delete the default group.

CHAPTER 3

Livelihood Functions in a Website

Livelihood WCM Server provides simple and comfortable methods for accessing special Livelihood objects, such as folders, documents, channels, and discussions, and for publishing these objects on a website. In the examples of this chapter, WCM-managed JSP pages are used to access the Livelihood objects. The Portal Manager API of Livelihood WCM Server provides Java beans for programming such JSP pages. The Portal Manager API is based on the WCM Java API which offers extensive functions for accessing Livelihood. This chapter provides application examples to illustrate the usage of these APIs.

The chapter deals with the following topics:

- preparations for using Livelihood functions in a website (see the following section)
- configuring multiple Livelihood systems (see section “Configuring Multiple Livelihood Systems” on page 86)
- overview of the Portal Manager API and WCM Java API functions for using Livelihood objects (see section “Overview of API Functions” on page 91)
- examples for programming with the Portal Manager API and the WCM Java API (see section “Application Examples” on page 95)

Preparations

Before you can use Livelink functions in your website, you must take the following steps:

1. Install the Livelink module “WCM Kernel Integration”, see the following section.
2. Activate the Livelink functions for the website, see “Activating the Livelink Functions for the Website” on page 83.
3. Add a Livelink pool, see “Adding a Livelink Pool” on page 84.
4. Assign the Livelink pool to the master Content server, see “Assigning the Livelink Pool to the Master Content Server” on page 85.

The individual steps are described in the following sections.

Installing the WCM Kernel Integration Module

The Livelink module “WCM Kernel Integration” enables Livelink WCM Server to access Livelink objects. It is also responsible for automatically updating relators (for which the *Update Edit version automatically* option is activated) when the associated Livelink object changes.

To install the module

1. The WCM Kernel Integration module is supplied as a file named **wcmobj.zip**. This file is located in the `\\livelink\` directory on the WCM CD. Extract the file to the directory `{Livelink installation directory}\staging\`.
2. Open the Livelink Administration page.

The URL is of type

<http://livelinkserver.company.example/Livelink/livelink.exe?func=admin.index>.

3. Log in to the Livelink system as the Admin user and go to the *Module Administration* section.

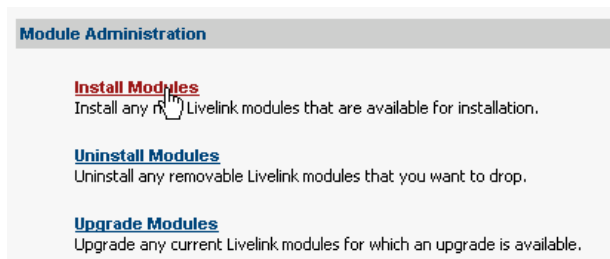


Fig. 29 – The *Module Administration* section on the Livelink Administration page

4. Click the *Install Modules* link.
5. On the Install Modules page, select the check box of the WCM Kernel Integration module and then click the *Install* button.

For further information on installing modules, refer to the Livelink online help.

To uninstall the module

In the *Module Administration* section of the Livelink Administration page, select the *Uninstall Modules* link. For further information on uninstalling modules, refer to the Livelink online help.

Activating the Livelink Functions for the Website

1. Launch the Admin client (see section “Starting the Admin client” on page 26).
2. Log in to the Admin client.
3. Select *Configuration* → *Websites* → {website name}.
4. Choose *Livelink extension* on the context menu.

5. Confirm the security prompt by clicking the Yes button.

On the *General* tab of the website settings, the check box *Website extension for Livelink* is selected.

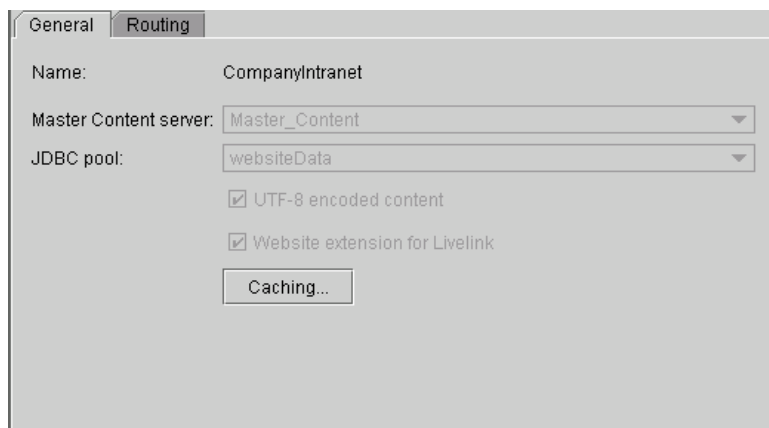


Fig. 30 – Website extension for Livelink

As a consequence, the two repositories `LivelinkObjectRepository` and `LivelinkUserRepository`, the `LivelinkApplication` application, and the “Livelink relator” object type are automatically created in the WCM system.

Adding a Livelink Pool

1. In the Admin client, select *Configuration* → *Pools* → *Livelink*.
2. Choose *New pool* on the context menu or click the corresponding icon.



Icon for adding a pool

3. The wizard for adding a pool starts.

The wizard is described in more detail in the online help of the Admin client.

Assigning the Livelink Pool to the Master Content Server

1. In the Admin client, select *Configuration* → *Servers* → {*name of the master Content server*} → *Pools*.
2. Choose *Assign pool* on the context menu or click the corresponding icon.



Icon for assigning a pool to a server

3. In the *Select pools* dialog box, mark the desired pool.



Fig. 31 – Assigning the Livelink pool to the master Content server

4. Confirm by clicking the *OK* button.

Configuring Multiple Livelink Systems

You can configure Livelink WCM Server to set up connections to several physical Livelink systems or connections to different entry points in the same Livelink system. The following steps are required:

1. Add Livelink pools.
2. Assign the Livelink pools to Content servers.
3. Shut down the WCM servers.
4. Modify the **livelink-systems.xml** file.
5. Start the WCM servers.

The individual steps are described in the following sections.

Adding Livelink pools

In the Admin client, add a separate Livelink pool for each Livelink system that you want to integrate and for each entry point respectively.

See “Adding a Livelink Pool” on page 84.

Assigning the Livelink pools to Content servers

Assign the individual Livelink pools to the corresponding Content servers.

See “Assigning the Livelink Pool to the Master Content Server” on page 85.

Shutting down all WCM servers

1. In the Admin client, select *System administration* → *Active servers*.
2. Choose *Set run level for all servers* on the context menu or click the corresponding icon.



Icon for setting the run level of all servers

The *Select new run level* window opens.

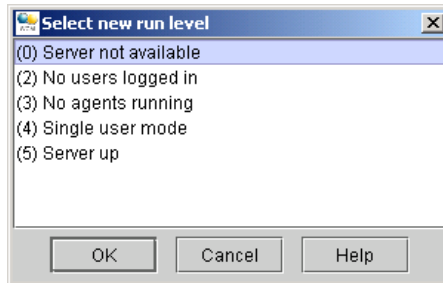


Fig. 32 – Selecting a new run level

3. Click the “Server not available” run level and confirm by clicking the *OK* button.

Modifying the livelink-systems.xml file

The **livelink-systems.xml** file is located in the directory **{WCM installation directory}\config**. For each Livelink system or entry point, add a separate **<livelink-system>** element to the configuration file.

livelink-systems.xml

```
<livelink-systems>

  <livelink-system name="Livelink_1">
    <default>true</default>
    <poolname>
      <default>livelink_1-pool</default>
      <server name="Proxy_server">proxy-livelink_1-pool</server>
    </poolname>
    <rootnodeid>2016091</rootnodeid>
    <url>http://livelink.company.example/Livelink</url>
    <llview>llview.exe</llview>
    <objectcache>
      <strategy>de.gauss.vip.[...].ObjectWiperImpl</strategy>
      <cache_reduce_interval>7200000</cache_reduce_interval>
      <class>de.gauss.vip.[...].DisplacingObjectCache</class>
      <max>2500</max>
    </objectcache>
  </livelink-system>
</livelink-systems>
```

```
        <min>2000</min>
        <wipe_strategy>time</wipe_strategy>
    </objectcache>
    <principalcache>
        <strategy>de.gauss.vip.[...].ObjectWiperImpl</strategy>
        <cache_reduce_interval>7200000</cache_reduce_interval>
        <class>de.gauss.vip.[...].DisplacingObjectCache</class>
        <max>20</max>
        <min>10</min>
        <wipe_strategy>time</wipe_strategy>
    </principalcache>
</livelink-system>

<livelink-system name="Livelink_2">
...
</livelink-system>

</livelink-systems>
```

Table 6 – Elements of the livelink-systems.xml

Element in the XML file	Explanation
<code><livelink-system name="Livelink_1"></code>	Name of the Livelink system
<code><default>true</default></code>	You can configure one Livelink system as the default Livelink system. The <code>LivelinkObjectBean</code> will initially use the default Livelink system.
<code><poolname></code>	
<code><default>livelink_1-pool</default></code>	
<code><server name="Proxy_server">proxy-livelink_1-pool</server></code>	
<code></poolname></code>	Name of the Livelink pool
	If more than one Content server accesses the Livelink system, you can configure a separate Livelink pool for each Content server.

Element in the XML file	Explanation
<code><rootnodeid>123456789</rootnodeid></code>	ID of the entry point into the Livelink system. This entry is optional. If you do not specify a root node ID, the Enterprise Workspace will be used as the starting point.
<code><url>http://livelink.company.example/Livelink</url></code>	URL of the Livelink system. This entry is optional.
<code><llview>llview.exe</llview></code>	Name of the viewer program used for the preview functionality in Livelink
<code><objectcache></code> <code> <strategy>de.gauss.vip.[...] .ObjectWiperImpl</strategy></code> <code> <cache_reduce_interval>7200000</cache_reduce_interval></code> <code> <class>de.gauss.vip.[...] .DisplacingObjectCache</class></code> <code> <max>2500</max></code> <code> <min>2000</min></code> <code> <wipe_strategy>time</wipe_strategy></code> <code></objectcache></code>	<p>Livelink objects can be loaded from the database into the object cache to speed up access to the objects.</p> <p><code><max></code>: Maximum number of objects in the cache</p> <p><code><min></code>: Number of objects to which the cache is reduced in server idle times</p> <p><code><cache_reduce_interval></code>: Interval in milliseconds. If the objects in the cache are not accessed during this period, objects are removed from the cache until the minimum cache size is reached again.</p>

Element in the XML file	Explanation
<pre><principalcache> <strategy>de.gauss.vip.[...].ObjectWiperImpl</strategy> <cache_reduce_interval>7200000</cache_reduce_interval> <class>de.gauss.vip.[...].DisplacingObjectCache</class> <max>20</max> <min>10</min> <wipe_strategy>time</wipe_strategy> </principalcache></pre>	<p>Principals can be loaded from the database into the cache to speed up access to the principals.</p> <p>This element has the same parameters as the <code><objectcache></code> element.</p>

Starting all WCM servers

1. Start the Admin server by means of the file **{Admin server name}.bat** or **{Admin server name}.sh** which is located in the WCM installation directory.
2. Use the file **startvip.bat** or **startvip.sh** to start the other WCM servers. This file is also located in the WCM installation directory. For starting WCM servers running in an application server, use the appropriate function of the application server.

Starting the servers in this way also starts up the websites.

The Livelink systems configured above are now available when you work in the Content client or write JSP pages based on the Portal Manager API or the WCM Java API.

Overview of API Functions

This section provides a basic overview of handling Livelink objects using the Portal Manager API or the WCM Java API. In order to learn how to write code based on the APIs, refer to section “Application Examples” starting on page 95.

For detailed information on how to use individual interfaces, beans, and methods for the integration with Livelink, refer to the Javadoc. The Javadoc documentation is located in the directory **{WCM installation directory}\documentation\javadoc**.

Portal Manager API

The Portal Manager API provides the following beans for integration with the Livelink:

- `de.gauss.vip.portalmanager.LivelinkObjectBean`

This is the main bean for accessing Livelink objects, such as documents, folders, discussions, and projects. Several create methods for adding objects directly in Livelink can be called via this bean.

- `de.gauss.vip.portalmanager.LivelinkUserBean`

This bean enables access to users and groups stored in Livelink.

The interface definitions for the actual Livelink objects are contained in the WCM Java API. The Portal Manager API gives you access to instances of these interfaces.

WCM Java API

Livelihood object types

The following table provides an overview of the supported Livelihood object types and their equivalents in the WCM Java API. As a rule, the interface names are built from the Livelihood term and the prefix “Livelihood”.

Table 7 – Livelihood object types and their equivalents in the WCM Java API

Livelihood object type	Interface in the WCM Java API	Package de.gauss.livelihood. ...
Alias	LivelihoodAlias	object
Document	LivelihoodDocument	object
Folder	LivelihoodFolder	object
Rendition	LivelihoodRendition	object
URL	LivelihoodURL	object
Version	LivelihoodVersion	object
Workspace	LivelihoodWorkspace	object
Channel	LivelihoodChannel	object.channel
News	LivelihoodNews	object.channel
Discussion	LivelihoodDiscussion	object.discussion
Topic	LivelihoodTopic	object.discussion
Reply	LivelihoodReply	object.discussion
Project	LivelihoodProject	object.project

Livelink object type	Interface in the WCM Java API	Package de.gauss.livelink. ...
Milestone	LivelinkMilestone	object.project
Task	LivelinkTask	object.project
Task list	LivelinkTaskList	object.project

The LivelinkObject interface constitutes the basis for the other interfaces. All interfaces mentioned above are directly or indirectly derived from LivelinkObject and thus inherit the methods of LivelinkObject.

Note: When you add Livelink objects using the LivelinkObjectBean of the Portal Manager API, the LivelinkId of the parent object must be specified. Please note that such a parent object must always be of type LivelinkFolder or must be derived from an object of this type.

Livelink users and groups

Livelink users and groups can also be accessed via the WCM Java API.

Table 8 – Livelink users and groups and their equivalents in the WCM Java API

Livelink term	Interface in the WCM Java API	Package de.gauss.livelink. ...
User	LivelinkUser	admin
Group	LivelinkGroup	admin

The LivelinkUser and LivelinkGroup interfaces are directly derived from the LivelinkPrincipal interface and thus inherit the methods of LivelinkPrincipal.

Livelink Permissions

In order to read or add Livelink objects, the user profile that is used for logging in must have the required permissions in Livelink. These are:

- “See Contents” for reading the metadata and the content of Livelink objects
- “See Contents” and “Add Items” for adding Livelink objects via the WCM APIs

For some Livelink object types which are similar to folders, such as discussions or channels, Livelink offers a simplified permission scheme. In this case, the rights “Read” or “Write” must be set for read or write access respectively.

Permissions when accessing several Livelink systems

You can configure more than one Livelink system for your WCM system. Each Livelink system manages separate user profiles which contain, among other things, the user permissions. When a user accesses an object, the system will only check the permissions which the user has for the connected Livelink system. It will not consider the user profile of the WCM system.

Application Examples

You can use WCM-managed JSP pages to access special Livelink objects, such as folders, documents, channels, and discussions, in a convenient fashion. This section provides application examples which illustrate how to use the Portal Manager API and the WCM Java API.

Note: The examples in this section show extracts from the technical concepts of the demo website “LiveStart”. The sample code cannot be executed as such; it is only provided for illustrating how to use the API functionality and how the JSP pages interact. The demo application “LiveStart” provides fully functional code. “LiveStart” is supplied as file **livestart-9.5.0.zip**. The ZIP file is located on the WCM CD in the **\examples** directory.

Integrating Content from a Livelink Folder in a Website

This application example illustrates how to display content and metadata of Livelink documents in your website. The permissions of the current user are implicitly evaluated. The example contains a simple login mechanism.

Scenario

The training department of your company manages all training materials in Livelink. You want to publish some of these materials on the website and provide personalized access to the documents.

Procedure

1. All documents contained in a Livelink folder are displayed in a list (**list.jsp**).
2. The user can log in to the system (**login.jsp**) and thus access additional documents.

3. Clicking a document name opens a new browser window displaying the content of the document (**show.jsp**).

Listing Documents of a Livelink Folder (list.jsp)

Using instances of the `LivelinkObject` interface, you can retrieve object information from the connected Livelink system. You need the node ID of the corresponding Livelink object for this.

The following sample code first creates an instance of `LivelinkObject`. The associated Livelink object with the node ID 42 is a folder. After this, selected metadata of the documents contained in this folder is output in a loop.

```
1      <jsp:useBean id="llobjBean" scope="session"
2          class="de.gauss.vip.portalmanager.LivelinkObjectBean" />
3      <%
4          // initialize bean
5          llobjBean.init(request);
6          // retrieve Livelink object instance
7          int nodeId = 42;
8          LivelinkObject llobj = llobjBean.get(new LivelinkId(nodeId));
9          if (llobj != null) {
10             // get children of Livelink object and iterate through them
11             List l = llobj.getChildrenIds();
12             for (int i=0; i<l.size(); i++) {
13                 // get an instance of the i-th document in the folder
14                 LivelinkId temp = (LivelinkId)l.get(i);
15                 LivelinkObject tempObj = llobjBean.get(temp);
16                 if (tempObj != null) {
17                     // extract metadata to show
18                     String comment = tempObj.getComment();
19                     LivelinkUser user = tempObj.getUser();
20                 }
21             }
22             %>
23             <b>name: <a href="show.jsp?nodeId=<%= temp.getNodeId() %>">
24                 <%= tempObj.getName() %></a></b><br />
25             <b>comment: <%= comment %> </b><br />
26             <b>created on:
27                 <%= tempObj.getAttribute(new StringValue("createdate")) %></b>
28                 <br/>
29             <b>created by: <%= user.getFirstName() %>
30                 <%= user.getLastName() %>
31                 (<%= user.getName() %>)</b><br />
32             <br />
```

```
26    <%  
27        }  
28    }  
29 }  
30 // Display login/logout button  
31 %>  
32 Logged in as:  
    <%= SessionBean.getCurrentUserProfile(request).getName() %>  
    <br/ >  
33 <a href="Login.jsp">login</login><br />
```

The node ID of the Livelink folder is specified directly in the code to simplify the example. You can also specify the node ID as a metadata item. The metadata item should be of type “Livelink ID”. You can use WCM tags to include metadata items in code (e.g. {VIPLIVELINKID format="nodeid"}).

User Login (login.jsp)

The SessionBean – a Portal Manager API class – associates an application server session with a context in the WCM system. The bean provides functions for logging users in to and out of the WCM system and for accessing user profiles.

The following sample first checks whether the user must log in. If so, the system displays the corresponding form for entering user ID and password. After the system has evaluated the login information, the JSP page is called again and the user is logged in. After login, the user will be directed to the page **list.jsp**.

```
1    <jsp:useBean id="sessionBean" scope="session"  
    class="de.gauss.vip.portalmanager.SessionBean" />  
2    <%  
3    // retrieve request parameter  
4    String user = request.getParameter("user");  
5    String pwd = request.getParameter("pwd");  
6    if (!sessionBean.isLoggedIn(request)  
    && user == null && pwd == null) {  
7    %>  
8    <form action="{VIPURL}" method="post">
```

```
9      Name: <input type="text" name="user"><br>
10      Password: <input type="password" name="pwd"><br>
11      <input type="submit">
12  </form>
13  <%
14  } else if (user != null && pwd != null)
15  {
16      sessionBean.checkLogin(request, user, pwd);
17  %>
18  <jsp:forward page="list.jsp" />
19  <%
20  }
21  %>
```

We recommend that you use the JSP tag `<jsp:forward>` cautiously, as this function is only available if no data has been returned to the browser yet. Alternately, you can use the following JavaScript command to force forwarding in the browser.

```
<script language="JavaScript">
    location.href="list.jsp";
</script>
```

Displaying the Content of Livelink Objects (show.jsp)

For displaying the content of a Livelink object, you can use the methods `hasContent` and `getContent` from the `LivelinkObject` interface. The node ID of the document to display is passed as a parameter to the JSP page. Based on the MIME type associated with the object, the system diverts the returned output stream.

```
1  <jsp:useBean id="llObjBean" scope="session"
2      class="de.gauss.vip.portalmanager.LivelinkObjectBean" />
3  <%
4  // initialize bean
5  llObjBean.init(request);
6  // Getting request parameter and instantiating Livelink object
7  String nodeStr = request.getParameter("nodeId");
8  if (nodeStr != null) {
9      int nodeId = Integer.parseInt(nodeStr);
```

```
9      LivelinkObject llobj = llObjBean.get(new LivelinkId(nodeId));
10     // if object is accessible and has a content show the content
11     if (llobj!=null && llobj.hasContent()) {
12         // retrieve mime type information
13         StringValue mime =
            (StringValue)llobj.getAttribute
            (LivelinkDocument.META_MIMETYPE);
14         // reset response stream and set mime type and status
15         response.reset();
16         if (mime != null)
17             response.setHeader("Content-Type", mime.getString());
18         response.setStatus
            (javax.servlet.http.HttpServletResponse.SC_OK);
19         // copying content to response stream
20         BufferedOutputStream sos = null;
21         try {
22             sos = new BufferedOutputStream
                (response.getOutputStream());
23             llobj.getContent(sos);
24         } catch(Exception e) {
25             if (sos!=null)
26                 sos.close();
27         }
28     }
29 }
30 %>
```

You can use the **show.jsp** page for displaying any Livelink document. Before the **show.jsp** page is executed, no data must be written to the OutputStream of the HttpServletResponse. For this reason, no template must be assigned to this JSP page. You also have to omit the normal header and body tags.

Implementing a Livelink Browser

This application example extends the previous example by displaying a recursive and personalized folder structure. It also provides a function for adding items in the Livelink folder.

Scenario

The training department of your company manages all training materials in Livelink. These documents are organized in a folder structure. You want to provide personalized access to these folders on the website. Also, you want to enable the Training staff to add new documents in the folders via the website.

Procedure

1. The example implements a navigation for the Livelink folder structure in an Explorer-like style. The documents within the current folder are displayed in a list (**browser.jsp**).

Clicking the document title displays the content of the document (see section “Displaying the Content of Livelink Objects (show.jsp)” on page 98).

2. On the **browser.jsp** page, the users can click a link referring to the **form.jsp** form in order to add documents to the current folder (**create.jsp**).

Navigating Recursive Livelink Folder Structures (browser.jsp)

The JSP page consists of two parts: One part displays the folder structure including a navigation, the other part displays a list of the documents in the current folder (see the following section). You can use a table, for instance, to place the two parts on the JSP page.

The JSP page receives the current Livelink folder in the request parameter `navNodeId`. First, the page retrieves the current folder's parent folder and generates a link to this parent folder. In this way, the users can navigate the folder structure. The link calls the JSP page again passing the node ID of the corresponding Livelink object in a parameter. The topmost folder, the Enterprise Workspace object, does not have a parent folder. Thus, no link is generated here. After this, links for the subordinate folders of the current object are generated into the JSP page.

```
1    ...
2    // determine node id
3    int navNodeId = 0;
4    String navNodeIdStr = request.getParameter("navNodeId");
5    if (navNodeIdStr != null) navNodeId =
        Integer.parseInt(navNodeIdStr);
6    if (navNodeId == 0) navNodeId = 2000;
7    // map storing all non-folder objects, used later for the list
8    RepositoryMap childrenMap = new RepositoryMap();
9    // retrieve the object and check whether the given object exists
10   LivelinkObject llobjNav =
        llobjBean.get(new LivelinkId(navNodeId));
11   if (llobjNav != null) {
12       // The navigation is not empty. First, print a ".." for the
13       // parent, if the parent exists.
14       LivelinkId parentId = llobjNav.getParentLivelinkId();
15       if (parentId != null) {
16           // Print ".." to link to the parent folder. This will call
17           //this JSP again, but with the node id of the parent.
18       %>
19       <b><a href="{VIPURL}?navNodeId=<%= parentId.getNodeId() %>">...
           </a></b>
20       <%
21       } // if
22       // Iterate the children, print their titles if they are folders
23       List lNav = llobjNav.getChildrenIds();
24       for (int i=0; i<lNav.size(); i++) {
25           // get child object
26           LivelinkId tId = (LivelinkId)lNav.get(i);
27           LivelinkObject tObj = llobjBean.get(tId);
28           // child exists and is a folder; create a link calling this
29           // JSP with the according node id
30           if (tObj != null && tObj.getCatalogMode() !=
               LivelinkObject.HIDDEN)
31       {
```

```
32             if (tObj.getObjectType().getSubType()==
33                 LivelinkObjectType.SUBTYPE_FOLDER) {
34         %>
35         <b><a href="{VIPURL}?navNodeId=<%=tempId.getNodeId() %>">
36             <%= title %></a></b>
37         <%
38             } else {
```

The childrenMap created in this process will be used for displaying the Livelink objects, which are not of type “folder”, in a list (see the following section).

```
37             // add element to the children map used in this page
38             childrenMap.putValue
39                 (temp, tempObj.getAsRepositoryEntry());
40         }
41     } // for
42 } // if
43 %>
```

Displaying Documents from a Livelink Folder (browser.jsp)

The second part of the JSP page displays a list of the documents which are contained in the Livelink folder. The documents are sorted by name in the list. Clicking a document name opens a new browser window showing the content of the document (see section “Displaying the Content of Livelink Objects (show.jsp)” on page 98).

The JSP page also offers a login method. After the user has logged in successfully, a link for adding items is shown.

JavaScript methods are used to open the browser window. The comment `/*VIPURL*/` causes the value which follows the comment to be treated as a WCM-managed link. In this way, you can use WCM-managed URLs in scripting code (JSP, JavaScript).

```
1    ...
2    <script language="JavaScript">
3    function openUploadWin() {
4        var uploadURL = /*VIPURL*/"form.jsp?uploadNodeId=
5        <%= navNodeId %>";
6        window.open( uploadURL ,
7            "name","width=650,height=400,left=200,top=150");
8    }
9    function openLoginWin() {
10        window.open(/*VIPURL*/"login.jsp",
11            "name","width=650,height=400,left=200,top=150");
12    }
13    </script>
```

The document list is displayed as a table and contains six metadata items for each Livelink object. You can easily modify the appearance of the list to suit your requirements.

The `isLoggedIn` method from the `SessionBean` class of the Portal Manager API checks whether the user is logged in to the system.

```
11    <table width="100%">
12    <tr>
13        <td colsapn="6" align="right">
14            <% // provide link depending on login status of the current session
15            if (SessionBean.isLoggedIn(request)) { %>
16                <a href="javascript:openUploadWin()">Add document</a>
17            <% } else { %>
18                <a href="javascript:openLoginWin()">login</a>
19            <% } %>
20        </td>
21    </tr>
22    <tr>
23        <td>Type</td><td>Title</td><td>Comment</td>
24        <td>User</td><td>Created</td><td>Modified</td>
25    </tr>
```

Sorting within a `RepositoryMap` is very flexible. Sorting always considers the attribute type. Thus, you can sort the entries by date values in ascending or descending order. The `RepositoryMap` can be also be sorted by several attributes. For more information, refer to the Javadoc of the `RepositoryMap` class.

```
26  <% // show all collected children, see above, sorted by name
27  RepositoryIterator iter = childrenMap.sort(LivelinkObject.NAME);
28  while (iter.hasNext()) {
29      RepositoryEntry entry = iter.nextEntry();
30      // retrieve some metadata of the current children
31      LivelinkId llid =
32          (LivelinkId)entry.getValue(LivelinkObject.META_LLID);
33      LivelinkObjectType type =
34          (LivelinkObjectType)entry.getValue
35              (LivelinkObject.META_OBJECTTYPE);
36      StringValue title =
37          (StringValue)entry.getValue(LivelinkObject.META_NAME);
38      StringValue comment =
39          (StringValue)entry.getValue(LivelinkObject.META_COMMENT);
40      LivelinkUser user =
41          (LivelinkUser)entry.getValue(LivelinkObject.META_USERID);
42      DateValue create =
43          (DateValue)entry.getValue(LivelinkObject.META_CREATEDATE);
44      DateValue modified =
45          (DateValue)entry.getValue(LivelinkObject.META_MODIFYDATE);
46      // generate a URL depending on objecttype
47      String url = null;
48      if (type.getSubType() == LivelinkObjectType.SUBTYPE_DOCUMENT)
49          url = /* VIPURL */"show.jsp?nodeId=" + llid.getNodeId();
50  }
51  %>
```

All Value classes of the Portal Manager API provide a `format` method for formatting the values of the class according to a locale and a format description. In the following example, the date is formatted according to the JSP page's locale and the `short` format.

```
43     <tr>
44         <td><%= type.format() %></td>
45         <% if (url != null) { %>
46             <td><a href="<%= url %>" target="content">
47                 <%= title.format() %></a></td>
48         <% } else { %>
49             <td><%= title %></td>
50         <% } %>
51         <td><%= comment.format(); %></td>
52         <td><%= user.getName() %></td>
53         <td><%= create.format(null, "short") %></td>
54         <td><%= modified.format(null, "short") %></td>
55     </tr>
56     <%
57 }
58 %>
</table>
```

LivelinkObject is used to retrieve the icon that is associated with the object. The URL used for accessing the icon can be composed with the following code:

```
String imgUrl = "{VIPDEPLOYMENT_URL}/" + livelinkObj.getImageUrl()
```

Displaying a Form for Adding a Document (form.jsp)

The following form can be provided for entering the information required for adding a Livelink document. The form contains input fields for the title, the comment, and for selecting a file. The node ID, which is passed on as a parameter, identifies the folder to which the document will be added. The **create.jsp** page that is called by **form.jsp** creates the document.

```
1     ...
2     String uploadNodeId = request.getParameter("uploadNodeId");
3     %>
4     <form name="form" action="create.jsp" method="post"
5         enctype="multipart/form-data" />
6         <input type="hidden" name="returl" value="browser.jsp" %>">
```

```
6      <table name="table" border="0" style="width:400px;"
7          align="center">
8          <tr><td>Title:</td>
9          <td><input type="text" name="title" style="width:100%">
10             </td></tr>
11          <tr><td>Comment:</td>
12          <td><input type="text" name="comment"></td></tr>
13          <tr><td>File:</td>
14          <td><input type="file" name="fileinput"></td></tr>
15      <%
16      int uId = Integer.parseInt(uploadNodeId);
17      LivelihoodObject llobj = llobjectBean.get(new LivelihoodId(uId));
18      if (llobj != null) {
19          %>
20          <tr><td>Folder</td>
21          <td><%= llobj.getName() %>
22              <input type="hidden" name="filetopic" value="
23                  <%= uploadNodeId %>">
24          </td></tr>
25      <%
26      }
27      %>
28      <tr>
29          <td colspan="2" align="center"><input type="submit"
30              value="Create">&nbsp;<input type="button" value="Close"
31              onClick="javascript:window.close()"></td>
32      </tr></table></form>
33      ...
```

Adding the Document (create.jsp)

The **create.jsp** page adds the Livelihood document to the folder. The page extracts all required parameters from the request object and adds the corresponding Livelihood document. The `ParseMultiFormData` class, which is part of the Portal Manager API, is used for extracting the data.

```
1      ...
2      OutputStream outS = null; InputStream inS = null;
3      LivelihoodObject llobj = null;
4      try {
5          // ParseMultiFormData is used to analyze the request of the
6          // field "input file".
```

```

7      ParseMultipartFormData parseMFD =
        new ParseMultipartFormData( request );
8      // Get the form data
9      String returnUrl = parseMFD.getParameterValues("returnl")[0];
10     String title = parseMFD.getParameterValues("title")[0];
11     String comment = parseMFD.getParameterValues("comment")[0];
12     String[] topicStr =
        parseMFD.getParameterValues("filetopic")[0];
13     int folderId = Integer.parseInt(topicStr);
14     // The content type is analyzed: Extract the file name with
15     // parseMFD, use java.io.File to get its name (without path and
16     // suffix)
17     File file =
        new File(parseMFD.getParameterValues("fileinput")[0]);
18     String fileName = file.getName();
19     // determine parent id
20     LivelinkId parent = new LivelinkId(folderId);

```

The WCM Java API provides functions for retrieving information about WCM servers and their configuration. Of course you can use the WCM Java API in conjunction with the Portal Manager API. In this example, the temporary directory of the installation is retrieved in order to store the file passed to the request.

```

21     // copy uploaded file to temp directory of current WCM server
22     String dirName =
        VipRuntime.getCurrentServer().getInstallDirectory() +
        File.separator +
        VipRuntime.getCurrentServer().getTempDirectory();
23     // creating tempfile in tempdir
24     file = new File(dirName, fileName);
25     // copy stream to file
26     inS = parseMFD.getFileContent(uploadParam , 0);
27     outS = new BufferedOutputStream(new FileOutputStream(file));
28     de.gauss.io.StreamConnector streamCon =
        new de.gauss.io.StreamConnector(inS, outS);
29     streamCon.copyAndClose();
30     // creating Livelink object
31     llobj =
        llObjectBean.createDocument(parent, title, comment, file);
32     file.delete();
33 }
34 catch {Exception e} {
35     if (inS != null) inS.close();

```

```
36     if (ourS != null) outS.close();
37   }
38   // if object could be created
39   if (llobj != null) {
40     %>
41     <script language="JavaScript">
42       location.href = '<%= returnUrl %>?navNodeId=
         <%= llobj.getLivelinkId().getNodeId() %>';
43     </script>
44     <% } else { %>
45     Error creating page<br />
46     <a href="#" onClick="javascript:history.back()">back</a>
47     <%
48   }
49   %>
```

Integrating a Livelink Channel in a Website

In Livelink, you can add and manage *News*. News are grouped to form a *Channel*. The Portal Manager API provides functions for accessing these special Livelink objects. You can create, read, and delete news and channels.

Channels

A channel is a special type of a Livelink folder. For this reason, the `LivelinkChannel` interface is derived from the `LivelinkFolder` interface. An instance of type `LivelinkChannel` contains a list of `LivelinkNews` objects. Two methods are available for retrieving news:

- `getActiveNews` returns a sorted list of `LivelinkID` instances. The list contains only those IDs whose news have not expired yet. The most current news item is on top of the list.
- `getNews` returns all – i.e. also expired – news items in an unsorted list of `LivelinkID` instances.

News

In order to retrieve information on a `LivelinkNews` object, you can use the methods `getHeadline`, `getStory`, `getUser`, `getImageId`, etc. from the `LivelinkNews` interface. The `getAttachments` method returns a list of attachments belonging to a news item.

The Javadoc contains a complete list of methods.

Use the `createChannel` and `createNews` methods from the `LivelinkObjectBean` to create new channels and news directly in the connected Livelink system.

Scenario

The Public Relations department of your company wants to publish press releases in the website. You want to display summaries of the current news in a teaser list as well as the full text of the news items.

Controlling the Page's Appearance (`show_channel.jsp`)

The page's appearance is defined by the `show_channel.jsp` page. The display mode of the JSP page is controlled by means of call parameters. The teaser list is displayed if no special node ID is passed to the JSP page.

```
1    ...
2    <%
3    // retrieving node ids from the current object and from the request
4    int nodeId = {VIPLIVELINKID format="nodeid"};
5    String currentNodeId = request.getParameter("nodeId");
6    // if no node id is passed, show the list
7    if (currentNodeId == null) {
8        // show list
9        ...
10   } else {
11       // show news
12       ...
13   }
```

14 ...

Displaying the Teaser List (show_channel.jsp)

The teaser list displays all current news sorted by “Valid from” date on the JSP page. Name, date, and comment are shown in a table.

```
1   ...
2   <h1>{VIPTITLE}</h1>
3   <table border="0" align="center" width="100%" cellpadding="0"
4       cellspacing="0">
5       <%
6       // show all active news from the related channel object
7       LivelinkChannel llChannel =
8           (LivelinkChannel)llObjBean.get(new LivelinkId(nodeId));
9       List l = llChannel.getActiveNews();
10      // iterate children
11      for (int i=0; i<l.size(); i++) {
12          // retrieving news and extract metadata to display
13          LivelinkId tempNewsId = (LivelinkId)l.get(i);
14          LivelinkNews tempNews =
15              (LivelinkNews)llObjBean.get(tempNewsId);
16          String comment = tempNews.getComment();
17          DateValue dateValue =
18              new DateValue(tempNews.getDateEffective());
19          %>
20          <tr><td><a href="{VIPURL}?nodeId=<%= tempNewsId.getNodeId() %>">
21              <%= tempNews.getName() %></a></td>
22              <td align="right"><%= dateValue.format(null, "medium") %>
23              </td></tr>
24          <tr><td colspan="2" style="text-align:justify;"><%= comment %>
25              </td></tr>
26          <tr><td colspan="2"><a href="{VIPURL}?nodeId=
27              <%= tempNewsId.getNodeId() %>">
28              See our news</a></td></tr>
29          <% } %>
30      </table>
31      ...
```

Displaying Single News Items (show_channel.jsp)

A single news item is displayed on a JSP page together with all attributes provided by Livelink. The icons and attachments of news are displayed by means of the JSP page **show.jsp** described above (see section “Displaying the Content of Livelink Objects (show.jsp)” on page 98).

```
1    ...
2    // retrieve news instance given through the request parameter
3    int nId = Integer.parseInt(currentNodeId);
4    LivelinkNews llCurrent =
        (LivelinkNews)llObjBean.get(new LivelinkId(nId));
5    // extracting metadata for display
6    String story = llCurrent.getStory();
7    // to format the news content, replace newlines with HTML tags.
8    story =
        de.gauss.util.StringOperator.replace(story, "\n", "<br />");
9    String headline = llCurrent.getHeadline();
10   LivelinkId imgId = llCurrent.getImageId();
11   String imgAlt = llCurrent.getImageAlt();
12   LivelinkUser user = llCurrent.getUser();
13   // show current entry
14   %>
15   <h1><%= llCurrent.getName() %></h1>
16   <h2><%= headline %></h2>
17   <table width="100%">
18   <tr><td width="80%" style="text-align:justify;"><%= story %></td>
19   <% // if an image is assigned to this news, show image
20   if (imgId!=null) {
21   %>
22       <td width="20%" valign="top">
23           " /></td>
25   <% } %>
26   </tr></table><table>
27   <% // show all attachments to this news instance
28   List a = llCurrent.getAttachments();
29   for (int i=0; i<a.size(); i++) {
30       LivelinkId aId = (LivelinkId)a.get(i);
31       LivelinkObject aObj = llObjBean.get(aId);
32       if (aObj!=null) {
33           String attImgUrl =
34               "{VIPDEPLOYMENT_URL}/" + aObj.getImageUrl();
35   %>
36   <tr><td>" /></td>
```

```
34      <td width="100%"><a href="show.jsp?nodeId=  
      <%= aObj.getLivelinkId().getNodeId() %>"  
      target="livelink_content"><%= aObj.getName() %></a></td>  
35      <td><%= aObj.getContentSize() %></td></tr>  
36      <% }  
37      } %>  
38      </table>  
39      ...
```

The procedure for adding news or channels in Livelink corresponds to adding documents (see section “Adding the Document (create.jsp)” on page 106). You write a form for entering the required information and create a JSP page which interprets the form data and adds the news items or channels via the `LivelinkObjectBean`. The interface for adding channels is `createChannel(parentId, title, comment, imageId)`; the interface for adding news is `createNews(parentId, title, comment, imageId, headline, story, effective, expired)`.

Integrating Livelink Discussions in a Website

Discussions belong to the frequently used Livelink features. Livelink discussions allow users to work together collaboratively to resolve issues and problems they encounter while working on projects. Thanks to the Portal Manager API, this Livelink feature can also be used in a website.

Livelink makes a distinction between the actual discussion object and the entries contained in the discussion object. The entries consist of *Topics* and *Replies* to the topics. The API provides methods for creating new topics and replies.

In this way, the Portal Manager API enables you to set up interactive discussion forums and to use the WCM functionality to apply your corporate design to the discussion.

Livelink discussions in websites can go well beyond the simple display of the discussions. You can also use the Portal Manager API functions to create *Frequently Asked Questions (FAQs)* from these discussions.

Scenario: Integrating Discussions in a Website

The company staff uses discussions to exchange information on general topics. These discussion forums are managed in Livelink. You want to publish some of these discussions in the intranet. The staff members shall be able to take part in the discussions via the intranet.

Procedure

1. All discussions from a specific Livelink area are shown (**find.jsp**).
The search method of the `LiveliinkObjectBean` is used for this purpose.
2. When the user selects a discussion, all topics of this discussion are displayed in a list (**topic.jsp**).
3. When the user selects a topic, the replies to this topic are shown hierarchically (**reply.jsp**). Additionally, the contents of the topic or reply is shown.
4. Buttons for adding new topics/replies are integrated in the topic and reply JSP page respectively (**form.jsp**). The **create.jsp** page creates these topics and replies.

The functions described here can also be implemented with fewer JSP pages (you will find an example for this in the LiveStart website). To make the application examples more comprehensible, the LiveStart code is not shown here.

Showing All Discussions of a Livelink Area (**find.jsp**)

The JSP page lists all available discussions which the logged-in user is allowed to view. The `LiveliinkObjectBean` is used to find the discussions. Then, the application iterates through the search results.

```
1    ...
2    <table>
3    <%
4    int llFilterId = {VIPLIVELINKID format="nodeid"};
```

```
5    // define filter
6    String filter = "(OTSubType :
    "+LivelinkObjectType.SUBTYPE_DISCUSSION+");";
7    if (llFilterId != 0)
8        filter += " AND (OTLocation : "+llFilterId+")";
9    // Search for all discussions and iterate through the result
10   RepositoryIterator iter = llObjBean.search(filter, true);
11   while (iter.hasNext()) {
12       // getting discussion object
13       LivelinkId temp = (LivelinkId)iter.nextKey();
14       LivelinkObject tempObj = llObjBean.get(temp);
15       // if you have access to the object, display the discussion
16       if (tempObj != null) {
17           // display node
18           String navImgUrl = "{VIPDEPLOYMENT_URL}/" +
               tempObj.getImageUrl();
19           String title = tempObj.getName();
20       %>
21       <tr><td valign="top"></td>
22       <td class="nav-topic"><a href="topic.jsp?discussionId=
               <%= temp.getNodeId() %>"><%= title %></a></td></tr>
23   <%
24       }
25   }
26   %>
27   </table>
28   ...
```

Displaying All Topics of a Selected Discussion (topic.jsp)

This page lists all topics of the selected Livelink discussion sorted by date. Below the topic list, a button for adding new topics is added to the JSP page.

```
1    ...
2    <table>
3    <%
4    String dId = request.getParameter("discussionId");
5    int discId = Integer.parseInt(dId);
6    LivelinkDiscussion llDiscussion =
        (LivelinkDiscussion)llObjBean.get(new LivelinkId(discId));
7    if (llDiscussion != null && llDiscussion.hasChildren()) {
```

```
8      // get all topics from the discussion
9      List l = llDiscussion.getTopics();
10     for (int i=0; i<l.size(); i++) {
11         LivelinkId tId = (LivelinkId)l.get(i);
12         LivelinkTopic tObj = (LivelinkTopic)llObjBean.get(tId);
13         String param = "discussionId=" + dId +
14             "&topicId=" tId.getNodeId() + "&nodeId=" tId.getNodeId();
15     <%
16     <tr><td><a href="reply.jsp?<%= param %>"><%= tObj.getName() %>
17         </td>
18         <td><%= tObj.getContent() %></td></tr>
19     }
20     }
21     <%>
22     </table>
23     <input type="button" name="Add Topic"
24         onClick="location.href='form.jsp?discussionId=<%= dId %>' />
25     ...
```

Displaying a Topic with all Replies (reply.jsp)

The JSP page displays the selected topic and lists all replies to this topic in a hierarchical structure. Below the hierarchical structure, a button for adding a reply is written to the page. The JSP page is called with the following parameters: the node IDs of the discussion, of the topic, and of the currently selected entry.

The `generateOut` method generates the hierarchical structure. The method traverses the structure of the passed `LivelinkDiscussionEntry` instance and generates a link for each element. Each element is indented according to its level in the hierarchy.

```
1     ...
2     <%!
3     /** Generate an entry for the tree. Will be called recursively. */
4     static String generateOut(LivelinkObjectBean llObjectBean,
5         LivelinkDiscussionEntry llObj, String param, int depth) {
6         // get required data for the given object
7         int width = depth*15; // 15 px per level indention
8         LivelinkId llId = llObj.getLivelinkId();
9         LivelinkUser user = llObj.getUser();
```

```
9      DateValue date = (DateValue)llObj.  
        getAttribute(LivelihoodObject.META_CREATEDATE);  
10     String title = llObj.getName() + " - " + user.getName() +  
        " - " + date.format(null, "short");  
11     String ret = "<span width='" + width + "'>&nbsp;</span>  
        <a href='{VIPURL}?' + i.param + "&nodeId=" +  
        llId.getNodeId() + "'>" + title + "</a><br />";  
12     // iterate through all replies and call this method.  
13     List l = llObj.getReplies();  
14     for (int i=0; i<l.size(); i++) {  
15         LivelihoodId id = (LivelihoodId)l.get(i);  
16         LivelihoodDiscussionEntry obj =  
            (LivelihoodDiscussionEntry)llObjectBean.get(id);  
17         if (obj != null) ret +=  
            generateOut (llObjectBean, obj, param, depth + 1);  
18     }  
19     return ret;  
20 } %>
```

The request parameters `discussionId`, `topicId`, and `nodeId` are used as parameters for the JSP page. `topicId` identifies the topic to be displayed. This topic also serves as the root element for the hierarchical structure. The `nodeId` is the currently selected discussion entry which is to be displayed. The `discussionId` is only required for further processing.

```
21 <%  
22 String dId = request.getParameter("discussionId");  
23 String tId = request.getParameter("topicId");  
24 String nId = request.getParameter("nodeId");  
25 String param = "discussionId=" + dId + "&topicId=" + tId;  
26 int topicId = Integer.parseInt(tId);  
27 LivelihoodTopic llTopic =  
    (LivelihoodTopic)llObjBean.get(new LivelihoodId(topicId));  
28 if (llTopic != null) {  
29     LivelihoodId nodeId = new LivelihoodId(Integer.parseInt(nId));  
30     LivelihoodDiscussionEntry entry =  
        (LivelihoodDiscussionEntry)llObjectBean.get(nodeId);  
31     // display the selected discussion entry, retrieve, manipulate  
32     // content, replace newline characters with line breaks.  
33     String content = entry.getContent();  
34     if (content.length()==0) content = "&nbsp;";  
35     content = de.gauss.util.StringOperator.replace  
        (content, "\n", "<br />");  
36     LivelihoodUser user = llCurrent.getUser();
```

```
37     DateValue date = (DateValue)entry.getAttribute
        (LivelinkObject.META_CREATEDATE);
38     %>
39     <h2><%= llCurrent.getName() %></h2>
40     <h3><%= user.getFirstName() %> <%= user.getLastName() %>
        (<%= user.getName() %>)</h3>
41     <p>created: <%= date.format(null, "medium") %></p>
42     <p><%= content %></p>
43     <hr />
```

A recursive call of the `generateOut` method generates the complete hierarchical structure. The `topicId` serves as the root element for the display. Below the hierarchical structure, a button for adding a reply is written to the JSP page.

```
44     <% // generate a tree for the current topic
45         out.println(generateOut(llObjBean, llTopic, param, 0);
46     }
47     // generate param to add a reply to the selected node
48     param += "&nodeId=" + nodeId;
49     %>
50     <input type="button" name="Add Reply"
        onClick="location.href='form.jsp?<%= param %>' />
51     ...
```

Displaying a Form for Adding a Topic or a Reply (form.jsp)

In this form, users can enter the information required for adding a topic or a reply. The form contains input fields for title and content. The node IDs of the discussion and a discussion entry are passed as parameters to the form. If no node ID for a discussion entry is passed, a new topic is created within the discussion. If a node ID is passed, a reply to the passed ID is created. The **create.jsp** page that is called by **form.jsp** creates the topic or reply.

```
1     // retrieve and check request parameter
2     String did = request.getParameter("discussionId");
```

```
3   String tId = request.getParameter("topicId");
4   String nId = request.getParameter("nodeId");
5   // get discussion entry
6   LivelinkId discId = new LivelinkId(Integer.parseInt(dId));
7   LivelinkObject discussion = llObjBean.get(discId);
8   if (discussion != null) {
9       String title = null;
10      int currentType = 0;
11      LivelinkDiscussionEntry parent = null;
12      // if the current node id is not set, a topic will be created,
13      // otherwise a reply
14      if (nId == null) {
15          title = "Post a New Topic";
16          currentType = LivelinkObjectType.SUBTYPE_TOPIC;
17      } else {
18          // get the current Livelink object
19          title = "Post a new Reply";
20          currentType = LivelinkObjectType.SUBTYPE_REPLY;
21      }
22      %>
```

The form calls the **create.jsp** page and passes the input fields for title and contents as well as the required node IDs and the object type (topic or reply) in hidden attributes.

```
23  <h1><%= title %></h1>
24  <form name="CreateItem" action="create.jsp" method="post">
25      <input type="hidden" value="<%= currentType %>"
26          name="objectType">
27      <input type="hidden" value="<%= dId %>" name="discussionId">
28      <input type="hidden" value="<%= tId %>" name="topicId">
29      <table cellSpacing="0" cellPadding="2" width="100%" border="0">
30          <tr><td>Subject:</td>
31          <td><input maxLength="248" size="60" name="Subject" />
32          </td></tr>
33          <tr><td>Comments:</td>
34          <td><textarea name="Comments" rows="12" cols="60">
35              </textarea></td></tr>
36          <tr><td align="center" >
37              <input type=submit value=Submit><input type=reset
38                  value=Reset></td></tr>
39      </table>
40  </form>
```

37 <% } %>

Adding a Topic or a Reply (create.jsp)

The last step in this example is to actually add the topic or reply object via the Portal Manager API. Adding these objects corresponds to the Livelink browser implementation (see section “Adding the Document (create.jsp)” on page 106). The request parameters are extracted from the request object. Depending on the passed object type, a topic or a reply is created. After this, the **reply.jsp** page is called which displays the new object.

```
1      // retrieve request parameter
2      String comment = request.getParameter("Comments");
3      String subject = request.getParameter("Subject");
4      String dId = request.getParameter("discussionId");
5      String tId = request.getParameter("topicId");
6      String nId = request.getParameter("nodeId");
7      String typeStr = request.getParameter("objectType");
8      if (typeStr!=null && dId!=null) {
9          int type = Integer.parseInt(typeStr);
10         LivelinkObject newObj = null;
11         if (type == LivelinkObjectType.SUBTYPE_TOPIC) {
12             LivelinkId parentId = new LivelinkId(Integer.parseInt(dId));
13             newObj = llObjBean.createTopic(parentId, subject, null,
14                 comment);
15         } else {
16             LivelinkId parentId = new LivelinkId(Integer.parseInt(nId));
17             newObj = llObjBean.createReply(parentId, subject, null,
18                 comment);
19         }
20         LivelinkId newLLId = newObj.getLivelinkId();
21         String url = "reply.jsp?discussionId=" + dId +
22             "&nodeId=" + newObj.getLivelinkId().getNodeId();
23         if (type == LivelinkObjectType.SUBTYPE_TOPIC)
24             url += "&topicId=" + newObj.getLivelinkId().getNodeId();
25         else
26             url += "&topicId=" + tId;
27     }
28     %>
29     <script language="JavaScript">
```

```
26 location.href="<%= url %>";
27 </script>
28 <%
29 }
```

Scenario: Providing Discussion Results as a FAQ

You want to use discussions that will not be changed anymore to publish FAQs in your intranet. The discussion topics are used to formulate questions, the corresponding answers are created on the basis of the discussion replies.

This example also shows the interaction between Java on the server side and JavaScript on the client side.

```
1 <h1>{VIPTITLE}</h1>
2 <p>Please select a question. The answer(s) will be listed on this
   page.</p>
3 <table border="0" width="100%" cellpadding="0" cellspacing="1">
4 <tr><td bgcolor="white">
5 <b>Questions:</b><br />
6 <% // determine the node id of the discussion object to show.
7 int nId = {VIPLIVELINKID format="nodeid"};
8 // getting discussion and retrieving questions
9 LivelinkDiscussion llDisc =
   (LivelinkDiscussion)llObjBean.get(new LivelinkId(nId));
10 // getting topics from the current discussion
11 List l = llDisc.getTopics();
12 StringBuffer str = new StringBuffer();
13 for (int i=0; i<l.size(); i++) {
14     // examine topic and generate question
15     LivelinkId tId = (LivelinkId)l.get(i);
16     LivelinkDiscussionEntry tObj =
       (LivelinkDiscussionEntry)llObjBean.get(tId);
17     if (tObj != null) {
18         String qContent = tObj.getContent();
19 %>
```

For each question, a <div> tag is written on the page that can be identified by its ID. The tag registers an EventListener that calls a JavaScript function when an element is clicked. This function switches the answer area on the client (see last code sample in this section). After this, an answers JavaScript array is generated which contains all answers and is used by the JavaScript method showAnswer. The array is embedded in the page (see last code sample in this section).

```
20  <div id="<%= tId.getNodeId() %>" onClick="showAnswer(this)">
    <%= qContent %></div>
21  <%    // adding all replies to the answer array
22      List ans = tObj.getReplies();
23      str.append("answers[\"\" + tId.getNodeId() + "\"] = \"\");
24      for (int j=0; j<ans.size(); j++) {
25          // retrieving reply
26          LivelinkId tReplyId = (LivelinkId)ans.get(j);
27          LivelinkDiscussionEntry tReply =
                (LivelinkDiscussionEntry)llObjBean.get(tReplyId);
28          if (tReply != null) {
29              String con = tReply.getName() + "</b><br />";
30              con += tReply.getContent();
31              con = de.gauss.util.StringOperator.replace(con,
                    "\n", "<br />");
32              str.append("<b>A: " + con + <br />");
33          }
34      }
35      str.append("\";");
36  }
37  }
38  %>
39  </td></tr></table>
40  <div id="question" style="display: none; visibility: hidden;">
    &nbsp;</div>
41  <div id="answer" style="display: none; visibility: hidden;">
    &nbsp;</div>
```

The last part of the code contains the JavaScript which switches the answers. After the composed array has been written on the page, the function for switching is defined. When the user clicks a question, this function is called. First, the DOM elements of the HTML page, which must be replaced, are retrieved and set to visible. After this, the node IDs of the question and the associated answer are retrieved based on the selected element. A DOM manipulation is performed to exchange the information on the page.

```
42  <script language="JavaScript">
43  <%= str.toString() %>
44  /** array with answer to given questions, will be initialized
    * during page creation */
45  var answers = new Array();
46  /** Shows the answer to the given question. */
47  function showAnswer(inItem) {
48      var question = document.getElementById("question");
49      var answerTable = document.getElementById("answer_table");
50      var answer = document.getElementById("answer");
51      if (answer && answerTable && question) {
52          question.style.display = "block";
53          question.style.visibility = "visible";
54          answer.style.display = "block";
55          answer.style.visibility = "visible";
56          var a = answers[inItem.id];
57          if (a == null || a.length == 0)
58              a = "Not answered yet!";
59          question.firstChild.nodeValue = "Q: " +
              inItem.firstChild.nodeValue;
60          answer.innerHTML = a;
61      }
62  }
63  </script>
```

Polling on the Website

In Livelink, you can take *Polls*. The Portal Manager API enables you to use these polls in a website.

You create the poll as usual in Livelink, i.e. you define the questions and alternative answers by means of the default Livelink functions. The Portal Manager API provides methods for retrieving these definitions. Also, users can take part in the poll (i.e. vote) and view the poll results.

A Livelink poll consists of one or more questions with predefined answers. There are two types of questions:

- question for which the users can choose exactly one of the predefined answers

In an HTML form, radio buttons are used for the answers.

- questions for which the users can choose as many answers as they like

In an HTML form, check boxes are used for the answers.

The conditions for voting in a website poll correspond to the conditions in Livelink:

- Users must log in before they can vote.
- Each user can only vote once.

Scenario

The staff of your company is to take part in a vote concerning “Safety on the job”. You want to make the poll available in the intranet. The staff members shall be able to vote via the intranet.

Procedure

1. The input form **form.jsp** provides functions for voting in a poll which already exists in Livelink.

2. Voting is performed on the **vote.jsp** page.
3. The **result.jsp** page displays the poll results.

Displaying a Form for a Livelink Poll (form.jsp)

The form must distinguish two cases:

- The user has not voted yet and can still do so.
- The user has voted already and can view the poll results.

The following example assumes that the user is already logged in to the system (see section “User Login (login.jsp)” on page 97). The `canVote(cid)` method makes the distinction between the two cases. The method checks whether the user has voted already and whether voting has not been completed yet.

```
1    <% // determine ContextId of the current user
2    ContextId cid = SessionBean.getCurrentContextId(request);
3    // load LivelinkPoll object (llId points to a LivelinkPoll object)
4    LivelinkPoll poll = (LivelinkPoll)livelinkObjectBean.get(llId);
5    if (poll != null) {
6        // user has the right to see and therefore vote the poll object
7        %>
8        <h1><%= poll.getName() %></h1>
9        <h2><%= poll.getComment() %></h2><hr />
10       <% // Check whether the user can vote
11           if (!poll.canVote(cid)) {
12               // user already voted or poll has been finished
13               %>
14               <h2>You already voted in this poll or
15                   the poll is finished already. You can
16                   <a href="result.jsp?llid=<%=llId.getNodeId()%>">
17                   view the poll results</a>.</h2>
18               <%
19                   } else {
20                       // show poll instructions and metadata
21                   %>
22               <h2>Instructions: <%=poll.getInstructions()%></h2>
```

The questions and answers of a poll must be displayed in the correct order, as the system only checks whether a predefined answer has been selected or not when evaluating the votes. The form only transfers the selected form elements. A corresponding index is embedded into the form elements in order to identify the assigned questions and answers on the **vote.jsp** page.

```
23 <form method="POST" action="vote.jsp">
24   <input name="llid" type="hidden" value="%=llid.getNodeId()%">
25   <table>
26   <% int questionIndex = 0;
27     for (Iterator qiter = poll.getQuestions().iterator();
28         qiter.hasNext(); questionIndex++) {
29       int choiceIndex = 0;
30       Question question = (Question)qiter.next();
31       %><tr><td colspan="2"><%=question.getQuestion()%"></td></tr><%
32         // decide to use radio button or check box controls
33         boolean radio = question instanceof AlternativesQuestion;
34         List choices = question.getChoices();
35         // show the choices
36         for (Iterator citer = choices.iterator(); citer.hasNext();
37             choiceIndex++) {
38       %><tr><td align="center">&nbsp;</td>
39       <td><input name="%=(\"Q\"+questionIndex)%" type="
40         <%= (radio? "radio": "checkbox") %>
41         value="%=(\"C\"+choiceIndex)%" /><%= (String)citer.next()%">
42         </td></tr>
43     } %>
44   <tr><td colspan="2">&nbsp;</td></tr>
45   <% } %>
46   </table>
47   <% // if comments are allowed add an input element for the comment
48     if (poll.isCommentAllowed()) {
49       %><textarea name="comment" cols="80" rows="5"></textarea>
50       <% } %>
51   <input type="submit" value="Vote">&nbsp;<input type="reset"
52     value="Reset">
53 </form>
```

Voting (vote.jsp)

Voting itself is performed on the **vote.jsp** page. The JSP page receives the answers to the questions as request parameters. The questions are identified by means of the parameter names and set accordingly in the poll.

```
1    <% // determine ContextId of the current user
2    ContextId cid = SessionBean.getCurrentContextId(request);
3    // extract form data from request
4    ParseFormEncoded pfe = new ParseFormEncoded(request);
5    // instantiate LivelinkPoll instance.
6    LivelinkId lllid = new LivelinkId
        (Integer.parseInt(pfe.getParameterValue("llid", 0)));
7    LivelinkPoll poll = (LivelinkPoll)livelinkObjectBean.get(lllid);
8    String comment = pfe.getParameterValue("comment", 0);
9    if(comment.length() < 1) comment = null;
10   for (Enumeration enum = pfe.getParameterNames();
        enum.hasMoreElements(); ) {
11       String parName = (String)enum.nextElement();
12       if(parName.startsWith("Q")) {
13           int questionIndex = Integer.parseInt(parName.substring(1));
14           // get all selected values for this
15           String questionString[] parValue =
                pfe.getParameterValues(parName);
16           for (int idx = 0; idx < parValue.length; idx++) {
17               int choiceIndex =
                    Integer.parseInt(parValue[idx].substring(1));
18               // select answer to the question. Type of the question
19               // must not be considered
20               poll.select(questionIndex, choiceIndex);
21           }
22       }
23   }
24   // send data to the server
25   poll.vote(cid, comment);
26   %>
27   <script language="JavaScript">
28       location.href = results.jsp?llid=<%=lllid.getNodeId()%>;
29   </script>
```

Displaying the Poll Results (result.jsp)

The current poll results can only be viewed if the user has voted already and if preview is explicitly allowed for the poll.

```
1    <% // determine ContextId of the current user
2    ContextId cid = SessionBean.getCurrentContextId(request);
3    // instantiate LivelinkPoll object
4    LivelinkId llid =
        new LivelinkId(Integer.parseInt(request.getParameter("llid")));
5    LivelinkPoll poll = (LivelinkPoll)lob.get(llid);
6    if (poll != null) {
7        // show name and comment
8        %>
9        <h1><%= poll.getName() %></h1>
10       <h2><%= poll.getComment() %></h2><hr />
11       <% // load poll results
12           LivelinkPollResults results = poll.getResults(cid);
13           int votingsTotal = results.getTotalNumberOfVotings();
14           DecimalFormat decfPerc = new DecimalFormat("#0.00");
15           %>
16       <h2>Total number of participants: <%=votingsTotal%></h2>
17       <h2>Instructions: <%=poll.getInstructions()%></h2>
18       <table>
19       <% // iterate through the questions
20           for (Iterator qiter = poll.getQuestions().iterator(),
21               riter = results.getResultsList().iterator();
22               qiter.hasNext(); ) {
21               Question question = (Question)qiter.next();
22               List choices = question.getChoices();
23               List votes = (List)riter.next();
24               %><tr><td colspan="4"><%=question.getQuestion()%></td></tr>
```

The questions and the corresponding answers of a poll are complemented by statistical information and illustrated by graphics.

```
25     <%   for (Iterator citer = choices.iterator(), viter =
26           votes.iterator(); citer.hasNext(); ) {
27               // determine statistical data
28               Integer numVotes = (Integer)viter.next();
29               double percent = (numVotes.doubleValue() * 100.0) /
30                   (double)votingsTotal;
31               int pix = numVotes.intValue();
32           %>
```

```
31 <tr><td align="right"><%= numVotes %>x</td>
32 <td><%= (String)citer.next() %></td>
33 <td align="right"><%= decfPerc.format(percent) %>%</td>
34 <td bgcolor="<%= (pix==votingsTotal?"#20DD10": "#E0E0E0") %>">
35 <table width="100"><tr>
36 <%
37         if(pix > 0 && pix < votingsTotal) {
38             pix = ((numVotes.intValue() * 100) + votingsTotal / 2)
39             / votingsTotal;
40             <td bgcolor="#20DD10" width="<%=pix%>">&nbsp;</td>
41 <%         } %>
42             <td>&nbsp;</td></tr></table></td></tr>
43 <%     }
44     } %>
45 </table>
```

Comments registered for the poll are listed at the end of the poll analysis.

```
46 <p>Total number of comments: <%= results.getNumberOfComments() %>
47 <% for (iter = results.getComments(0, -1); iter.hasNext(); ) {
48     %>
49 <p><%= iter.next() %>
50 <% } %>
```

CHAPTER 4

Livelihood Contents in the Content Client

Note: For detailed information on working with the Content client, refer to the Content Client User Manual.

Due to the integration of the WCM system with Livelihood, you can add Livelihood objects as WCM objects to a WCM-managed website. These WCM objects are called *relators*. They always refer to a certain version of the Livelihood object. The WCM object type “Livelihood relator” enables you to integrate a single Livelihood object in a website; the WCM object type “Livelihood folder relator” is provided for integrating Livelihood folders.

Note: The Livelihood module “WCM Kernel Integration” is required to access Livelihood objects via the WCM system. This module is also responsible for automatically updating the WCM object when the associated Livelihood object changes. For information regarding the installation of this module, refer to “Installing the WCM Kernel Integration Module” on page 82.

You can directly access relators in the WCM system (e.g. with the Content client). The actual objects, however, are stored in the Livelihood database. Relators are treated like normal WCM objects. They are subject to the staging of Livelihood WCM Server. That means there are three views of a relator (Edit, QA, and Production). The reference management of Livelihood WCM Server is also available for relators. Livelihood folder relators

have a special feature: As they do not have any content, you cannot check out and check in Livelink folder relators.

Livelink documents integrated in a website by means of relators are displayed directly in the browser. If files cannot be displayed directly in the browser, e.g. PDF files or Word documents, a so-called surrogate page containing a link to the file is generated. Livelink folders become WCM topics. As WCM topics have content, an HTML page is generated for each Livelink folder.

In the Navigation area of the Content client, the following icons symbolize relators:



The letters "LL" in the bottom right corner of the object type icon stand for Livelink. (The figure shows the relator icon for an Excel document).



If the system cannot access the Livelink object referenced by the relator, the relator's icon is marked by a red cross. The reason for this may be that the Livelink object has been deleted or that the connection between the WCM system and the Livelink system has been interrupted.

This chapter deals with the following topics:

- adding relators (see section "Adding Relators" on page 133)
- editing relators (see section "Editing Relators" on page 149)
- deleting relators (see section "Deleting a Relator" on page 163)

Adding Relators

There are different ways of adding Livelink objects as WCM objects. You can:

- add a single Livelink document as a WCM object (see the following section)
- add an existing Livelink folder structure as WCM objects (see section “Adding a Folder Relator” on page 139)
- create a Livelink document together with the associated relator in the Content client (see section “Adding a Relator Together with the Associated Livelink Document” on page 144)

Notes:

You cannot use the Content client to add folders in Livelink.

You should not **manually** add relators and folder relators below a folder relator. The process that synchronizes the WCM system with the Livelink system checks whether all the subfolders and documents in the Livelink folder are represented by relators; manually created relators are deleted.

Prerequisites for adding relators

- In the WCM system, you have the access rights “Read” and “Create” for the topic to which the relator is to be added.

If you want to add a new Livelink document together with the relator, you also need the permission “Add Items” for the Livelink folder to which you want to add the object.

- You are assigned the functional area “Livelink”. To check your functional areas, choose *Tools* → *Login info* in the Content client.

- Your WCM administrator prepared the website, in which you want to create the relator, for using Livelink functionality (see section “Preparations” on page 82).

Adding a Relator

Objective

You want to integrate a single Livelink object in a WCM-managed website. A wizard guides you through the process of adding the relator.

Starting the wizard

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the topic to which you want to add the relator.
5. Choose *Object* → *New* → *Relator*.

Specifying the Livelink system

There may be more than one Livelink system in your company. In the first dialog box, you specify the Livelink system in which the Livelink object you want to integrate is stored.

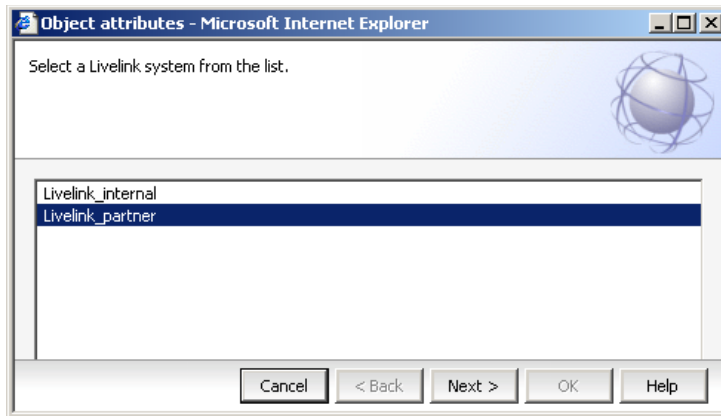


Fig. 33 – Specifying the Livelink system

Confirm by clicking the *Next* button.

Selecting a Livelink object

In this dialog box, you select the Livelink object that is to be added as a relator in the WCM system.

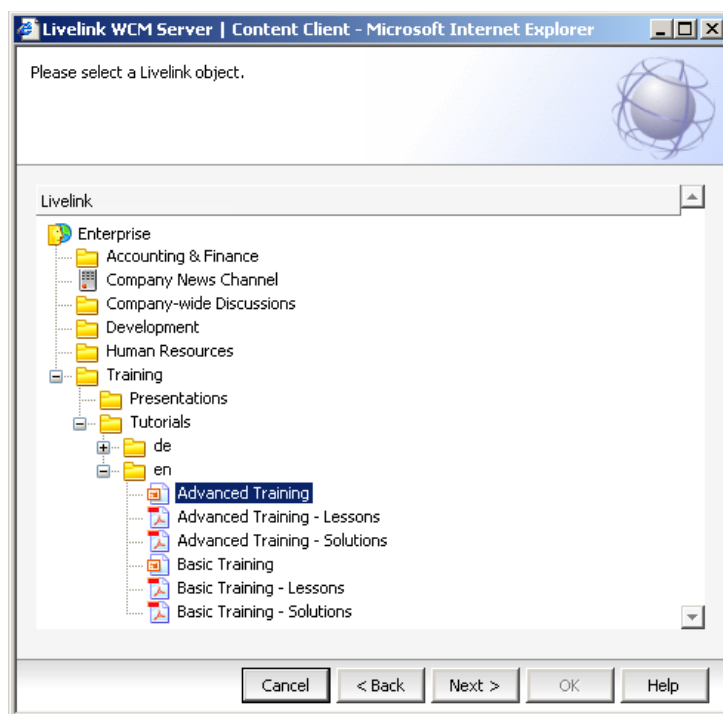


Fig. 34 – Selecting the Livelink object when adding a relator

Confirm by clicking the *Next* button.

Specifying the version of the Livelink object

The Livelink object that the relator refers to may be subject to change. In this dialog box, you determine how the relator reacts when the Livelink object changes.

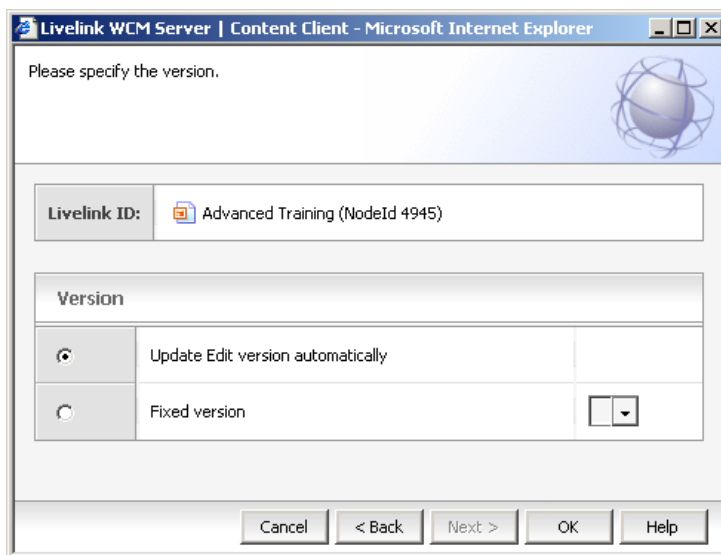


Fig. 35 – Specifying the version of the Livelink object

- **Livelink ID:** Livelink ID of the selected object. The Livelink ID is composed of the name and the ID of the object in the Livelink system. You cannot change this entry.
- **Update Edit version automatically:** If you click this radio button, the Edit view of the relator will be automatically changed whenever the object is modified in the Livelink system. For the changes to become visible in the published website, you must submit the updated relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Note: A relator can only be updated automatically if its status is either “changed”, “rejected”, “released”, or “delayed release”. If this requirement is not met, use the appropriate staging action to change the relator's status. After this, you can adapt the relator to the modified Livelink object by choosing *Object* → *Synchronize*.

- *Fixed version*: If you click this radio button, the relator will be based on a certain version of the Livelink object. When the object changes in the Livelink system, the relator remains unchanged. You can, however, update the relator manually via the Livelink metadata dialog box (see section “Modifying Livelink Metadata” on page 155).
- *Version*: If you clicked the *Fixed version* radio button, select the desired version of the Livelink object from the drop-down list.

Confirm your entries by clicking the *OK* button.

Entering a remark

In this dialog box, you may enter information for the relator's log.

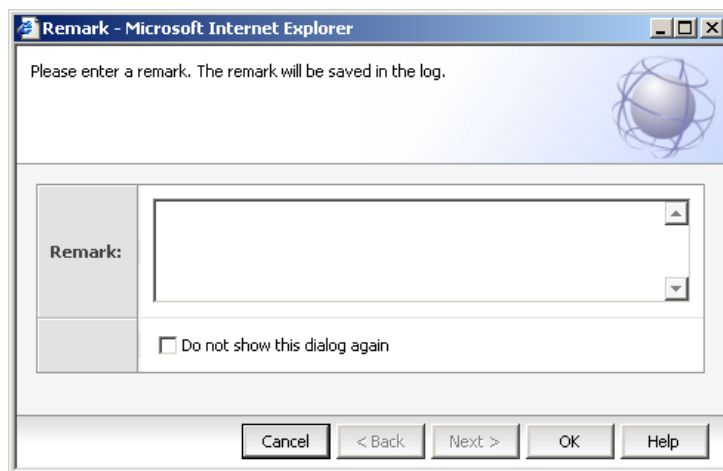


Fig. 36 – Entering comments about the relator

Confirm your entries by clicking the *OK* button.

The WCM system creates the relator and shows it in the Navigation area. The relator's status is “changed”.

For the relator to become visible in the published website, you must submit it. After this, the relator must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Adding a Folder Relator

Objective

You want to integrate a Livelihood folder including all subfolders and subordinate documents into the WCM system. Each Livelihood object is to be represented by a relator in the WCM system. A wizard guides you through this task.

Starting the wizard

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the topic in which you want to integrate the Livelihood objects.
5. Choose *Object* → *New* → *Relator*.

Specifying the Livelink system

There may be more than one Livelink system in your company. In the first dialog box, you specify the Livelink system in which the Livelink folder you want to integrate is stored.

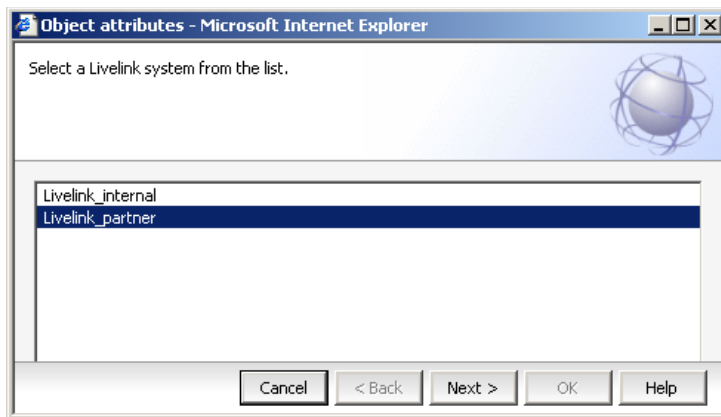


Fig. 37 – Specifying the Livelink system

Selecting the Livelink folder

In this dialog box, you select a Livelink folder. The subfolders and documents contained in this folder will be integrated in the WCM system.

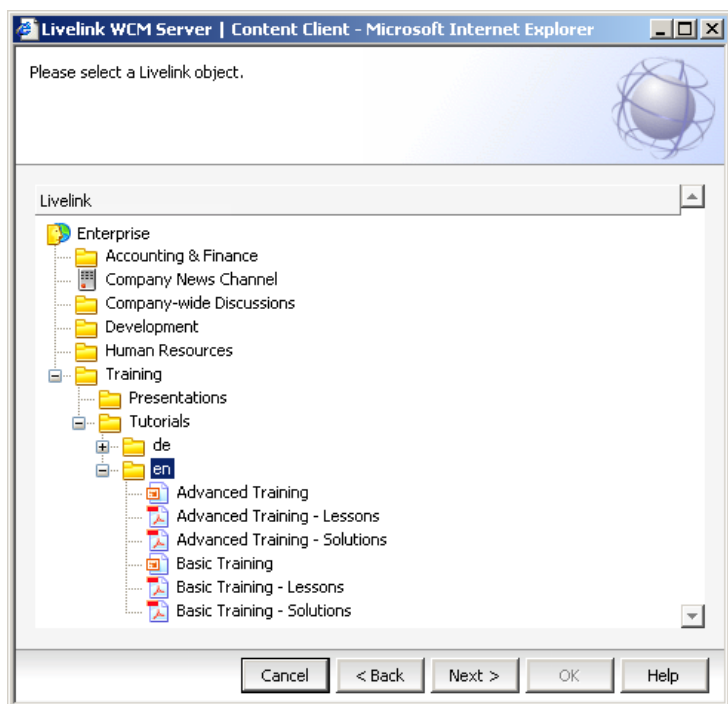


Fig. 38 – Selecting the Livelink folder

Confirm by clicking the *Next* button.

Specifying the version of the Livelink folder

The Livelink folder that the Livelink folder relator refers to may be subject to change. In this dialog box, you determine how the Livelink folder relator reacts when the Livelink folder changes.

Note: For each object in the Livelink folder, a separate relator is created. The setting that you make here initially applies to all relators. However, you can subsequently change this setting for individual relators (see section “Modifying Livelink Metadata” on page 155).

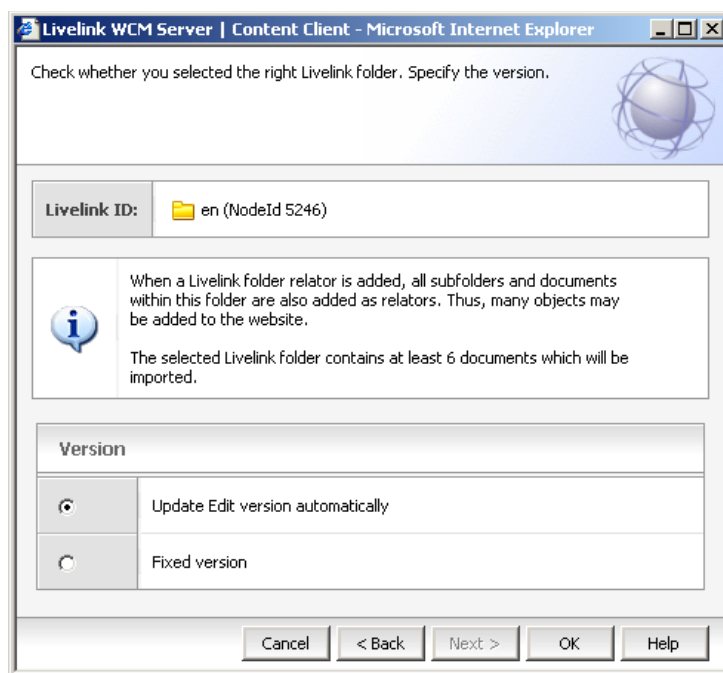


Fig. 39 – Specifying the version of the Livelink folder relator

- **Livelink ID:** Livelink ID of the selected folder. The Livelink ID is composed of the name and the ID of the folder in the Livelink system. You cannot change this entry.
- **Update Edit version automatically:** If you click this radio button, the Edit view of the Livelink folder relator will be automatically adapted whenever the metadata of the associated Livelink folder change. If an object is added to the Livelink folder, a new relator representing this object is automatically created. If an object is removed from the Livelink folder, the associated relator will be deleted.

For the changes to become visible in the published website, you must submit the changed relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Note: A relator can only be updated automatically if its status is either “changed”, “rejected”, “released”, or “delayed release”. If this requirement is not met, use the appropriate staging action to change the relator's status. After this, you can adapt the relator to the modified Livelink object by choosing *Object* → *Synchronize*.

- **Fixed version:** If you click this radio button, the Livelink folder relator will be based on the current version of the Livelink folder. When the folder's metadata change in the Livelink system, the associated Livelink folder relator remains unchanged. If an object is added to the Livelink folder, a new relator representing this object is not automatically created. However, the relator for an object will be deleted, if the object is removed from the Livelink folder.

You can adapt the relator manually via the Livelink metadata dialog box (see section “Modifying Livelink Metadata” on page 155).

Confirm your entries by clicking the *OK* button.

Entering a remark

In the following dialog box, you may enter information for the logs of the individual relators.

Confirm your entries by clicking the *OK* button.

The WCM system creates the relators and shows them in the Navigation area. Their status is “changed”.

For the relators to become visible in the published website, you must submit them. After this, the relators must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Note: Below a Livelink folder relator, you can add further WCM objects. If you selected the *Update Edit version automatically* option for the Livelink folder relator, you must consider the following: The Livelink folder relator **and** all subordinate WCM objects will be deleted automatically, when the referenced Livelink folder is deleted.

Adding a Relator Together with the Associated Livelink Document

Objective

You want to integrate a relator in the WCM system which refers to a file which does not yet exist as a document in the Livelink system. Together with the relator, you want to add this file as a new document in Livelink. A wizard guides you through this process.

Note: You cannot use the Content client to add folders in Livelink.

Starting the wizard

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the topic to which you want to add the relator.
5. Choose *Object* → *New* → *from object type* or click the appropriate icon.



Icon for adding an object on the basis of the object type

Specifying title, file, and object type

In the first dialog box, you specify the file which is to be added to the Livelink system and the name of the new Livelink document.

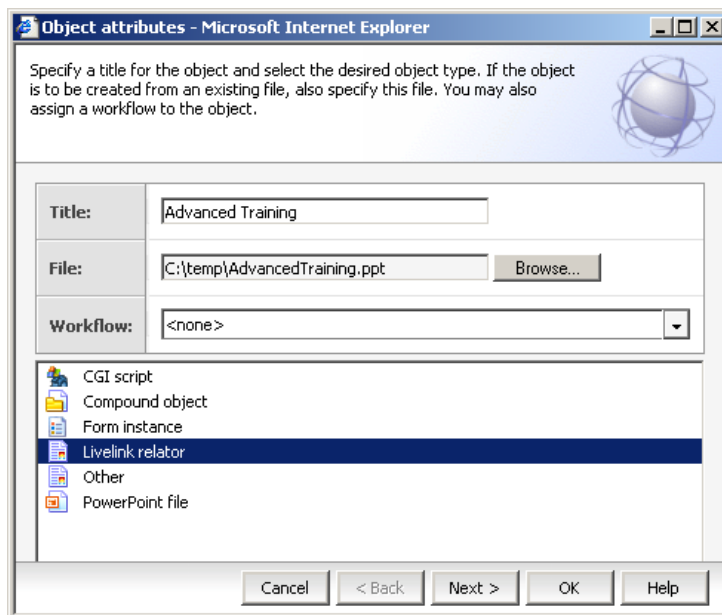


Fig. 40 – Specifying title, file, and object type

- **Title:** Enter the name for the document in Livelink. The Content client uses this name as the relator's title. The title is used to identify the relator and is displayed both in the header and in the Navigation area of the Content client.

Entering a title is mandatory.

Note: Titles must not be more than 255 characters long. Also, you should adapt the length of the title in such a way that the resulting file name meets the requirements of your operating system. Under Windows, the absolute path to a topic and a generated page may not be longer than 240 and 260 characters respectively.

- *File:* Specify the document to be added to the Livelink system. Click the *Browse* button to select the file.

Specifying a file is mandatory.

Note: Please choose only files that have a file extension.

- *Workflow:* If you want to assign a workflow to the new object, click the desired workflow in the drop-down list. For more information about the workflow of the WCM system, refer to the Content Client User Manual (chapter “Content Workflow”).
- *Object type:* Select the object type “Livelink relator”.

Confirm by clicking the *Next* button.

Specifying the Livelink system

There may be more than one Livelink system in your company. In the following dialog box, you select the Livelink system to which the document will be added.

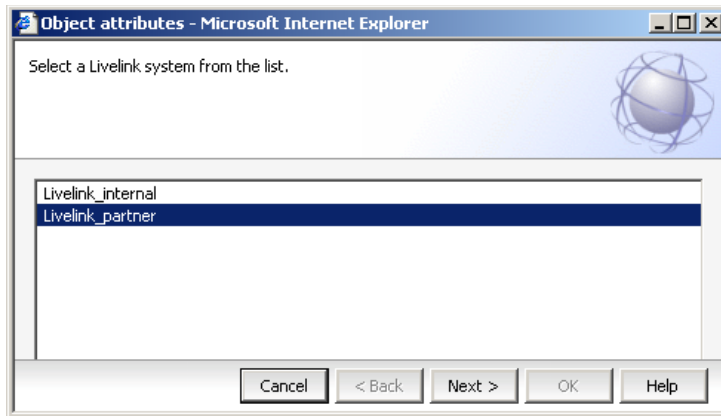


Fig. 41 – Specifying the Livelink system

Confirm by clicking the *Next* button.

Specifying location in Livelink

In the following dialog box, you select the Livelink folder for the new document.

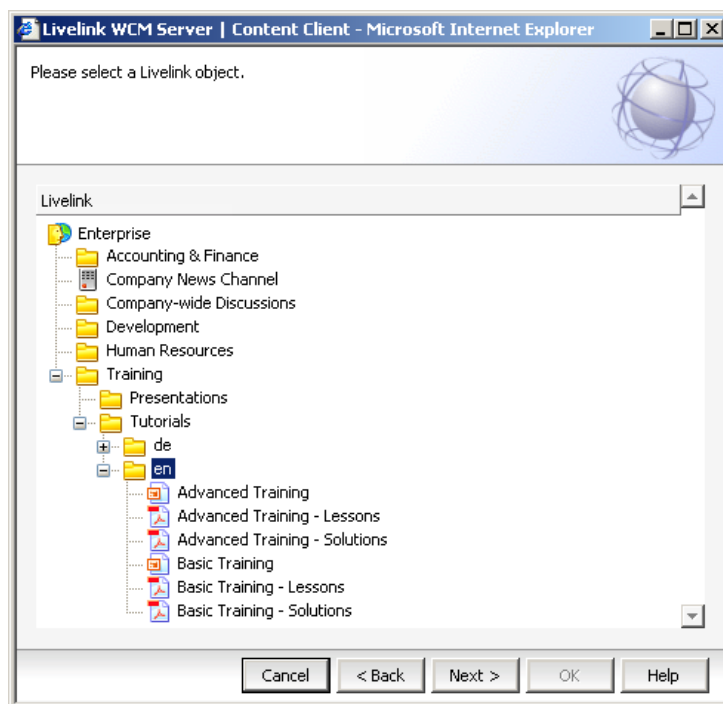


Fig. 42 – Selecting the Livelink folder

Confirm by clicking the *Next* button.

Entering a remark

In the following dialog box, you may enter information for the relator's log.

Confirm your entries by clicking the *OK* button.

The WCM system creates the relator and shows it in the Navigation area. The relator's status is "changed". Also, the document will be stored at the specified location in the Livelink system.

The relator is created with the settings *Update Edit version automatically* and *Delete Livelink object together with relator*. You can change these settings in the Livelink metadata dialog box (see section “Modifying Livelink Metadata” on page 155).

For the relator to become visible in the published website, you must submit it. After this, the relator must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Editing Relators

Relators do not only enable you to publish Livelink objects on a website. They also make it possible to edit Livelink objects.

In this section, the following procedures are introduced:

- submitting and releasing changed relators (see the following section)
- editing the content of a relator (see section “Modifying the Content of a Relator” on page 154)
- editing the Livelink metadata (see section “Modifying Livelink Metadata” on page 155)
- integrating the Livelink metadata in a web page (see section “Integrating Livelink Metadata in a Page” on page 159)
- manually updating the relator when the Livelink object has been changed (see section “Updating a Relator Manually” on page 162)

Submitting and Releasing Relators

Changes to relators only become visible in the published website after the relators were submitted and released by Quality Assurance.

Finding Changed Relators

You want to find all changed relators which can be submitted to Quality Assurance. This also includes relators that have been updated automatically because the referenced Livelink objects were modified. You can use the predefined filter “changed” for finding changed objects.

Requirement

You are assigned the functional area “Filter standard”. To check your functional areas, choose *Tools* → *Login info* in the Content client.

Procedure

1. Launch the Content client.

The URL of the Content client is of type
<http://wcmserver.company.example/cmsclient/index.jsp>
2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the topic in which you want to search.
5. Choose *View* → *Object filter*.
6. Click the “changed” filter in the drop-down list.

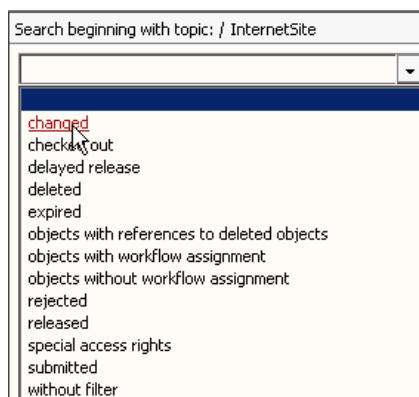


Fig. 43 – Object filter in the Content client

The filter hitlist opens automatically. It contains all changed WCM objects, not only relators.

Type	OID	Title	Version	Status
	4	Advanced Training	0.0.1	
	5	Advanced Training - Lessons	0.0.1	
	6	Advanced Training - Solutions	0.0.1	
	2	Basic Training	0.0.1	
	7	Basic Training - Lessons	0.0.1	
	3	Basic Training - Solutions	0.0.1	
	1	InternetSite	0.0.1	
	8	Presentations	0.0.1	

Fig. 44 – Filter results

You can select one or more objects in the list in order to edit them.

Submitting Relators

You have added a relator, changed an existing relator, or the relator has been updated automatically because the underlying Livelink object was modified. Quality Assurance is to check the relator. That means that you want to submit the relator.

Prerequisites

- The relator has the status “released”, “changed”, or “rejected”.
- You have the access rights “Read” and “Change object” for the relator.

Procedure

1. Launch the Content client.

The URL of the Content client is of type
<http://wcmserver.company.example/cmsclient/index.jsp>
2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the relator that you want to submit to Quality Assurance.
5. Choose *Object* → *Submit* or click the appropriate icon.



Icon for submitting an object

The object has the status “submitted” and can be checked by Quality Assurance.

Releasing Relators

You have completed the quality assurance process for a relator and come to a positive conclusion. You now want to transfer the current version of the relator to the Production view and thus make it available to the public.

Prerequisites

- The relator has the status “submitted”.
- You have the access rights “Read” and “Release” for the relator.

Procedure

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. In the Navigation area, mark the relator that you want to release.
4. Choose *Object* → *Release* or click the appropriate icon.



Icon for releasing an object

The changed relator is now visible in the published website.

Modifying the Content of a Relator

Objective

Relators refer to objects in the Livelink system. You can use the Content client to modify the content of Livelink objects.

Prerequisites

- The relator has the status “changed”, “rejected”, “released”, or “delayed release”.
- The option *Update Edit version automatically* is selected for the relator.
- You have the access rights “Read” and “Change object” for the relator.
- You have the permission “Reserve” for the Livelink object.

Procedure

1. Launch the Content client.

The URL of the Content client is of type
<http://wcmserver.company.example/cmsclient/index.jsp>
2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the relator whose associated Livelink object you want to edit.
5. Choose *Edit* → *Check out* or click the appropriate icon.



Icon for checking out an object

The relator is barred for write access by other users; the Livelihood object will be marked as reserved in the Livelihood system.

Depending on the configuration of the Content client, the content of the Livelihood object opens in the integrated HTML editor or in the associated editor program.

6. Edit the Livelihood object in the respective editor.
7. Save your changes and close the editor.
8. To return the modified relator to the WCM system and unreserve the Livelihood object, choose *Edit* → *Check in* or click the appropriate icon.



Icon for checking in an object

The relator has the status “changed”. A new version is added to the Livelihood object.

For the changes to become visible in the published website, you must submit the modified relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Modifying Livelihood Metadata

Objective

Each Livelihood object has a number of metadata. That means that relators have Livelihood metadata in addition to the WCM metadata, such as Livelihood system and Livelihood ID. You can view and modify the Livelihood metadata of relators in the Livelihood metadata dialog box.

You want to edit the Livelihood metadata of a relator.

Prerequisites

- The relator has the status “changed”, “rejected”, “released”, or “delayed release”.
- You have the access rights “Read” and “Change metadata” for the relator.
- You are assigned the functional area “Livelihood”. To check your functional areas, choose *Tools* → *Login info* in the Content client.
- You have the permission “Modify” for the Livelihood object.

Procedure

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the relator whose Livelihood metadata you want to edit.
5. Choose *View* → *Livelihood metadata*.

General Livelink data		Training Materials	
Livelink system:	Livelink_internal		
Livelink ID:	Advanced Training (NodeId 4945)		
Version:	1		
Created:	Jun 2, 2004 2:04:12 PM	Created by:	Admin
File name:	OnlineHelp2.ppt	MIME Type:	application/vnd.ms-powerpoint
Object size:	971.50 KB		
Description:	<div></div>		
Version description:	<div></div>		
Version	<input checked="" type="radio"/> Update Edit version automatically <input type="radio"/> Fixed version		
	<input type="checkbox"/> Delete Livelink object together with relator		
<div>Save</div> <div>Reset</div> <div>Help</div>			

Fig. 45 – The Livelink metadata dialog box

The dialog provides several tabs for viewing and editing metadata.

- **General Livelink data** tab: Most entries on this tab are controlled by the Livelink system and can only be viewed here.

In the fields *Description* and *Version description*, you may enter basic information on the relator and on this special version of the relator.

In the *Version* area, you determine whether the relator is to be updated automatically when the associated Livelink object changes (*Update Edit version automatically* radio button) or is to reference a specific version of the Livelink object (*Fixed version* radio button).

Select the *Delete Livelink object together with relator* check box to determine that the referenced Livelink object will be destroyed together with the relator (see section “Deleting a Relator” on page 163).

- Tabs for object categories: For each object category assigned to the object in the Livelink system, a separate tab with the category's name is shown (in the figure *Training Materials*). You can modify the Livelink object's attributes on the tabs.

6. Make the desired changes on the tabs.

7. Click the **Save** button.

The metadata are changed in the Livelink system. The relator now has the status “changed”.

For the changes to become visible in the published website, you must submit the modified relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Integrating Livelink Metadata in a Page

Just like WCM metadata, you can integrate Livelink metadata in a template or in the page generated for the object and thus use the metadata in a WCM-managed website. For integrating metadata, WCM tags are provided.

Note: For detailed information on WCM tags, refer to the Content Client User Manual (chapter “Working with WCM Tags”).

Livelink Standard Metadata

The following table provides an overview of the WCM tags for the most common Livelink standard metadata. The file **Livelink_WCM_tags.html** in the directory **{WCM installation directory}\examples\wcmtags** contains all Livelink-specific WCM tags.

Table 9 – WCM tags for the most common Livelink metadata

Metadata item	Internal name	WCM tag
Description	comment	{VIPLL attribute="comment"}
File name	filename	{VIPLL attribute="filename"}
Created by	createdby	{VIPLL attribute="createdby"}
Created	createdate	{VIPLL attribute="createdate"}
Modified	modifydate	{VIPLL attribute="modifydate"}
MIME type	mimetype	{VIPLL attribute="mimetype"}
Object size (in byte)	filedatasize	{VIPLL attribute="filedatasize"}

Metadata item	Internal name	WCM tag
Node ID	nodeid	{VIPLL attribute="nodeid"}
Version	versionnum	{VIPLL attribute="versionnum"}
Version description	vcomment	{VIPLL attribute="vcomment"}

User-Defined Metadata

The WCM tag mechanism is also available for user-defined Livelink metadata (e.g. the attributes of an object category). Use the following pattern to form WCM tags for user-defined attributes:

```
{VIPLL category="<category name>" set="<set name>" attribute="<attribute name>" set_no="<number>" item_no="<number>"}
```

- `category`: name of the category containing the attribute
- `set`: name of the set containing the attribute
- `attribute`: name of the attribute for which the WCM tag is created
- `set_no`: position of the set in the list of set values
- `item_no`: position of the attribute value to be shown in the list of attribute values

If the attribute to be shown is not of type “set”, the expression becomes simpler:

```
{VIPLL category="<category name>" attribute="<attribute name>" item_no="<number>"}
```

Example

For the training materials, you created a category named “Training Materials”. By means of the attributes of this category, you can manage information on the status and the contact person for the materials. The attribute set “Contact Information” of the category contains three attributes in the following order:

1. Contact Person
2. E-mail
3. Phone number

The attribute “Phone number” may contain two values: in the first position the conventional phone number, in the second position the mobile phone number.

The screenshot shows a web interface for managing content. At the top, there are tabs: General, Specific, Audit, Categories, Ratings, References, and Versions. The 'Categories' tab is selected. Below the tabs, there's a section for 'Training Materials' with a warning icon. Under this, there are two dropdown menus: 'Level' set to 'Advanced' and 'Status' set to 'In progress'. Below these is a yellow-highlighted section titled 'Contact Information'. This section contains three input fields: 'Contact Person' with the value 'Jstein', 'E-Mail' with the value 'Jstein@company.example', and 'Phone number' with two rows of values: '+1 555 666 777' and '+1 898 999 333'. At the bottom of the form are buttons for 'Submit', 'Apply', 'Reset', 'Remove', and 'Cancel'.

Fig. 46 – The category “Training Materials” in the Livelink system

Use the WCM tag

```
{VIPLL category="Training Materials" set="Contact Information"
attribute="Phone number" set_no="1" item_no="2"}
```

to show the mobile phone number of the contact person on the page. In case of the employee “Joshua Stein” who is entered as the contact person, this is the phone number “+1 898 999 333”.

Updating a Relator Manually

Objective

If the *Fixed version* option is selected for a relator, this relator will **not** be updated automatically when the referenced Livelink object changes. You want to update the relator manually.

Prerequisites

- The relator has the status “changed”, “rejected”, “released”, or “delayed release”.
- The *Fixed version* option is selected for the relator.
- You have the access rights “Read” and “Change metadata” for the relator.
- You are assigned the functional area “Livelink”. To check your functional areas, choose *Tools* → *Login info*.

Procedure

1. Launch the Content client.

The URL of the Content client is of type
<http://wcmserver.company.example/cmsclient/index.jsp>
2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. In the Navigation area, mark the relator that you want to update.

5. Choose *View* → *Livelink metadata* → *General Livelink data* tab.
6. In the *Version* drop-down list, click the version of the Livelink object that the relator is to reflect.
7. Click the *Save* button.

For the changes to become visible in the published website, you must submit the modified relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

Deleting a Relator

Notes

- You can delete relators like all other WCM objects. For detailed information on deleting WCM objects, refer to the Content Client User Manual (chapter “Editing Objects”).
- **Delete Livelink object together with the relator:** In order to destroy the Livelink object, you require the Livelink permission “Delete”. If you do not have this permission, the relator will be deleted in the WCM system. However, the object in the Livelink system will not be removed.
- **Livelink relators** are always deleted when the referenced objects are deleted in Livelink. Relators that have the status “submitted” are automatically rejected and deleted afterwards. If a relator has already been released once, it is marked as deleted and must be destroyed manually.
- **Livelink folder relators** for which the *Update Edit version automatically* option is selected, are automatically deleted (including all subordinate objects), when the referenced Livelink folder is deleted.

If you want to add further WCM objects below the Livelink folder relator, you should click the *Fixed version* radio button for the folder relator. Folders with this settings will not be removed when the referenced Livelink folder is deleted.

Selecting the *Delete Livelink object together with relator* option

In the Livelink metadata dialog box, you can determine that the referenced Livelink object will be destroyed together with the relator.

Proceed as follows:

1. Launch the Content client.

The URL of the Content client is of type

<http://wcmserver.company.example/cmsclient/index.jsp>

2. Log in to the Content client and select the website that you want to edit.
3. Change to the Edit view of the website by choosing *Tools* → *View* → *Edit view*.
4. Open the Livelink metadata dialog box by choosing *View* → *Livelink metadata*.
5. On the *General Livelink data* tab, select the check box *Delete Livelink object together with relator*.
6. Click the *Save* button.

For the change to take effect, you must submit the relator which then must be released by Quality Assurance (see section “Submitting and Releasing Relators” on page 150).

CHAPTER 5

Livelihood Search in the WCM System

The Search Server Connector for Livelihood enables you to use the search engine integrated in Livelihood for searching WCM-managed websites. Several scenarios are possible.

- Scenario 1 – Searching website contents

Livelihood is able to index contents from websites which are managed by the WCM system. For this purpose, an internal collection is added in the WCM system which forms the basis for searching the contents in a WCM-managed website.

- Scenario 2 – Searching Livelihood contents

An existing Livelihood slice can be added as an external collection in the WCM system. In this way, you can search Livelihood contents in a WCM-managed website.

- Scenario 3 – Searching other document sets

You can also index documents which are neither managed with Livelihood nor with a WCM system. The external collection created in the indexing process can be searched in a WCM-managed website.

This chapter deals with the following topics:

- technical concepts of the search engine integration, see the following section
- setting up the Livelink search for a WCM system, see section “Configuring the Livelink Search for the WCM System” on page 171
- permissions for displaying search results, see section “Permissions for Searching” on page 203
- using the Livelink search engine in a website, see section “Using the Search” on page 204
- managing external collections with command line tools, see section “Command Line Tools” on page 205

This section is intended for administrators with the necessary knowledge of the search functions provided by Livelink WCM Server.

- what to do in case of errors, see section “Problems and Solutions” on page 215

Technical Concepts

Livelink is able to index documents from external systems (in this case from the WCM system). The external data source is defined by an XML Activator within the Search Server Connector. This XML Activator comprises the following processes:

1. Process monitoring the directory in which external programs store the documents to be indexed

The files stored in this directory contain the actual content as well as the metadata of the documents. The most important metadata item is the URL, which is also called OTURN (Open Text Universal Resource Name) in this context.

2. Conversion process

The XML Activator includes a text converter supporting several file formats, such as DOC and PDF.

3. Indexing process

The documents converted to text are indexed and can thus be searched.

After converting and indexing have been completed, a new slice is defined on the basis of the data source. This slice contains the data generated during indexing. A search manager is created for processing the search requests.

Data flows during indexing

In order to use the Livelink search in a WCM system, you have to perform several installation and configuration steps (see section “Configuring the Livelink Search for the WCM System” on page 171). After this, the Livelink search engine regards the WCM-managed websites as external data sources.

Whenever a WCM object changes in one of the collections managed by the Livelink Search server, the ContentMinerNotification agent initiates the generation of a command file for the XML Activator. In this way, the XML Activator receives instructions on how to process the changes, e.g. re-index the object or remove the object from the index. The command file is created in a temporary directory and then moved to the directory monitored by the XML Activator.

System architecture

The following figure illustrates the interaction between the different components.

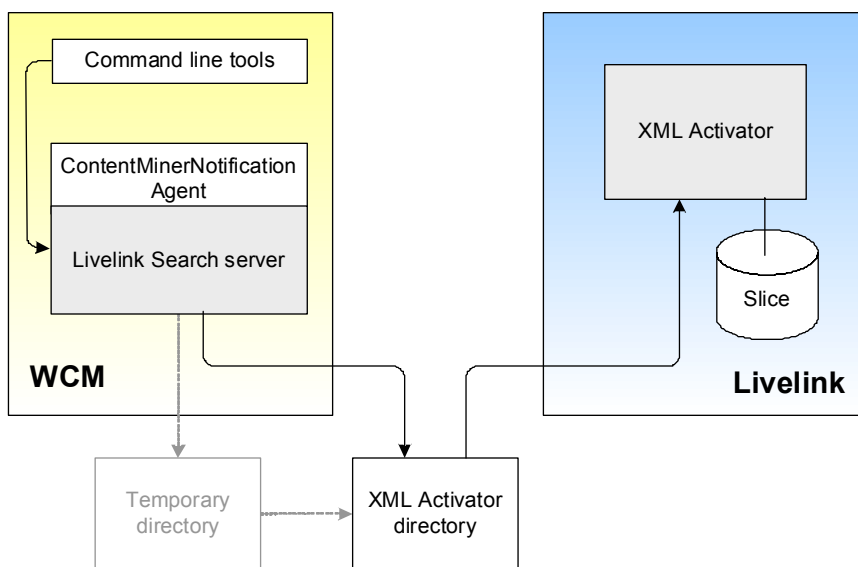


Fig. 47 – Interaction of components in the Livelink search

To make the illustration more comprehensible, only one XML Activator is shown. You can, however, configure several XML Activators. That means there may exist more than one:

- temporary directory
- XML Activator directory
- XML Activator
- slice

The relation between the individual components is always as shown in the illustration.

Configuring the Livelink Search for the WCM System

The procedure for setting up the Livelink search for the WCM system depends on the Livelink Enterprise Server version used in your company. If you use

- Livelink Enterprise Server version 9.2.1, read the following section
- Livelink Enterprise Server version 9.5.0, refer to section “Search Based on Livelink Enterprise Server 9.5.0” starting on page 188

Search Based on Livelink Enterprise Server 9.2.1

Note: Livelink and the WCM system use a shared directory for exchanging data. For this reason, both the Livelink system and the WCM server, to which the Search server is assigned, must have write access to this shared directory.

The Livelink search can be used in different scenarios.

Scenario 1 – Searching website contents

Realizing this scenario requires the following steps in Livelink and the WCM system:

1. WCM system: Install Livelink Search server (see the following section).
2. Livelink system: Prepare indexing (see section “Preparing Indexing” on page 177).
3. Livelink system: Create XML Activator Producer data flow (see section “Creating an XML Activator Producer Data Flow in Livelink” on page 178).

4. WCM system: Add collections (see section “Adding an internal collection for WCM contents” on page 182).
5. Livelink system: Make index regions queryable (see section “Making Index Regions Queryable in Livelink” on page 186).
6. Livelink system: Start the XML Activator process (see section “Starting the XML Activator Process in Livelink” on page 186).
7. Livelink system: Display index regions (see section “Displaying Index Regions in Livelink” on page 187).

Scenario 2 – Searching Livelink contents

Realizing this scenario requires the following steps in the WCM system:

1. Install Livelink Search server (see the following section).
2. Add collections (see section “Adding an external collection for a Livelink slice” on page 184).

Note: The Livelink module “WCM Kernel Integration” is required to access Livelink objects via the WCM system. For information regarding the installation of this module, refer to “Installing the WCM Kernel Integration Module” on page 82.

Scenario 3 – Searching other document sets

Realizing this scenario requires the following steps in Livelink and the WCM system:

1. WCM system: Install Livelink Search server (see the following section).
2. Livelink system: Prepare indexing (see section “Preparing Indexing” on page 177).
3. Livelink system: Create XML Activator Producer data flow (see section “Creating an XML Activator Producer Data Flow in Livelink” on page 178).

4. WCM system: Add collections (see section “Adding an external collection for another type of document set” on page 185).
5. Livelink system: Start the XML Activator process (see section “Starting the XML Activator Process in Livelink” on page 186).

Installing Livelink Search Servers in the WCM System

Note: For detailed information on installing a WCM system, refer to the Livelink WCM Server Installation Manual.

During the installation of a WCM system, the dialog box *Search server* opens. On the *Livelink* tab of this dialog box, you can add the desired number of Livelink Search servers.

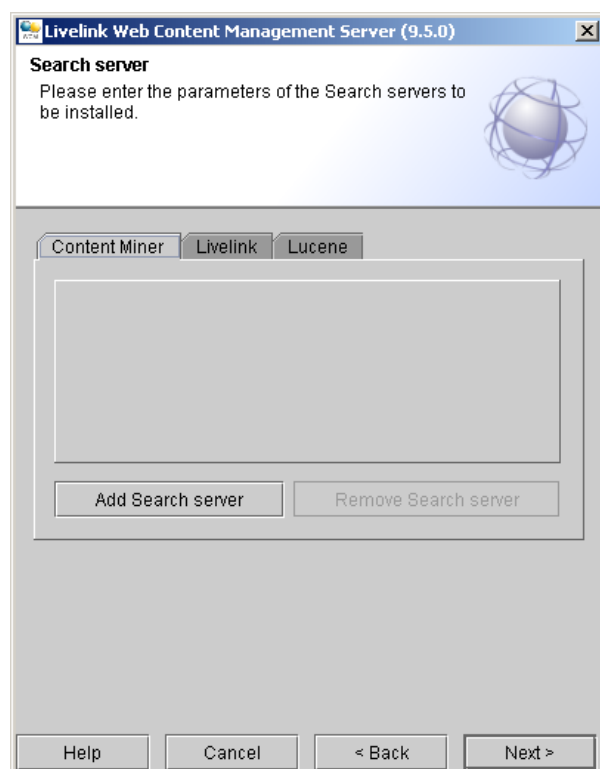


Fig. 48 – Search servers in the WCM installation program

1. Click the *Livelihood* tab.
2. Click the *Add Search server* button.

This opens a tab on which you can make the settings for the Livelihood Search server.

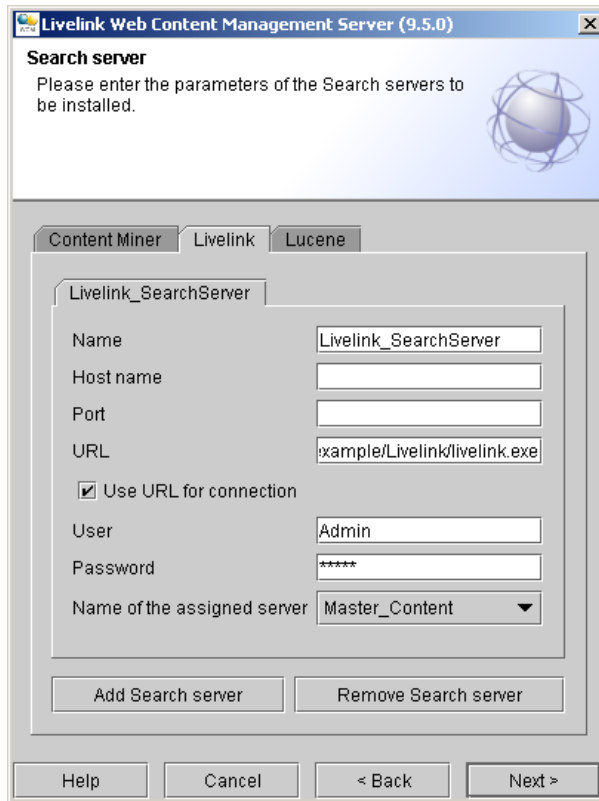


Fig. 49 – Installing Livelink Search servers

- **Name:** name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted: a–z, A–Z, 0–9 (ASCII 7 Bit), . (dot), - (hyphen) and _ (underline).
- **Host name:** fully qualified name of the computer hosting the Livelink server which is to perform the indexing and to process the search requests
- **Port:** connection port on the Livelink server (default: 2099)

- **URL:** URL to the Livelink server which is to perform the indexing and to process the search requests. The URL is of type <http://livelinkserver.company.example/Livelink/livelink.exe>.
- **Use URL for connection:** Select this check box if you want to use the Livelink server's URL for the connection. In this case, the entries in the fields *Host name* and *Port* will be ignored.
- **User:** user for processing the search requests in the Livelink system. The user must have read access to the slices.

Notes:

For the Livelink search, you can determine which index regions are queryable and displayable (see sections “Making Index Regions Queryable in Livelink” on page 186 and “Displaying Index Regions in Livelink” on page 187). These settings are user-specific. For this reason, we recommend that you configure a special user for the Livelink search. Enter the information of this user here.

For further information about permissions for the Livelink search, refer to section “Permissions for Searching” on page 203.

- **Password:** password for processing search requests in the Livelink system
 - **Name of the assigned server:** the name of a WCM server that already exists or is to be installed and to which you want to assign this Search server
3. When you have completed all settings for the Search servers, confirm the entries by clicking the *Next* button.

Preparing Indexing

Step 1 – Update the xmlactiv program

For the Livelink Search server to index the data correctly, the most recent version of the xmlactiv program must be executed in your Livelink system.

Copy the xmlactiv program from the directory

\\livelinksearch\{operating system}\bin on the WCM CD to the directory **{Livelink installation directory}\bin**.

Step 2 – Add WCM regions to the xml.tok file

Add the following region definitions to the file **{Livelink Livelink installation directory}\config\xml.tok**. The entries must be inserted directly below the section “Basic Open Text region definitions”.

```
#
# WCM Region Definitions for Livelink Index
#
REGION "VIPoid" FROM "<VIPoid" TO "</VIPoid>";
REGION "VIPwebsite" FROM "<VIPwebsite" TO "</VIPwebsite>";
REGION "VIPdeploymentsystem" FROM "<VIPdeploymentsystem" TO
    "</VIPdeploymentsystem>";
REGION "VIPcollection" FROM "<VIPcollection" TO "</VIPcollection>";
REGION "VIPtitle" FROM "<VIPtitle" TO "</VIPtitle>";
REGION "VIPsubtitle" FROM "<VIPsubtitle" TO "</VIPsubtitle>";
REGION "VIPkeywordlist" FROM "<VIPkeywordlist" TO "</VIPkeywordlist>";
REGION "VIPdescription" FROM "<VIPdescription" TO "</VIPdescription>";
REGION "VIPurl" FROM "<VIPurl" TO "</VIPurl>";
REGION "VIPpath" FROM "<VIPpath" TO "</VIPpath>";
```

You will find a file named **xml.tok** containing these entries in the directory **{WCM installation directory}\examples\livelinksearch**.

All entries for a REGION must be located in a single line. The regions VIPoid and VIPcollection are mandatory. The other regions can be used for extended search functions.

Creating an XML Activator Producer Data Flow in Livelink

You create an XML Activator Producer data flow to index information generated by a third-party application, such as the WCM system.

To create an XML Activator Producer data flow in the Livelink system:

1. Open the Livelink Administration page.

The URL is of type

<http://livelinkserver.company.example/Livelink/livelink.exe?func=admin.index>.

2. Log in to the Livelink system as the Admin user and navigate to the *Search Administration* section.

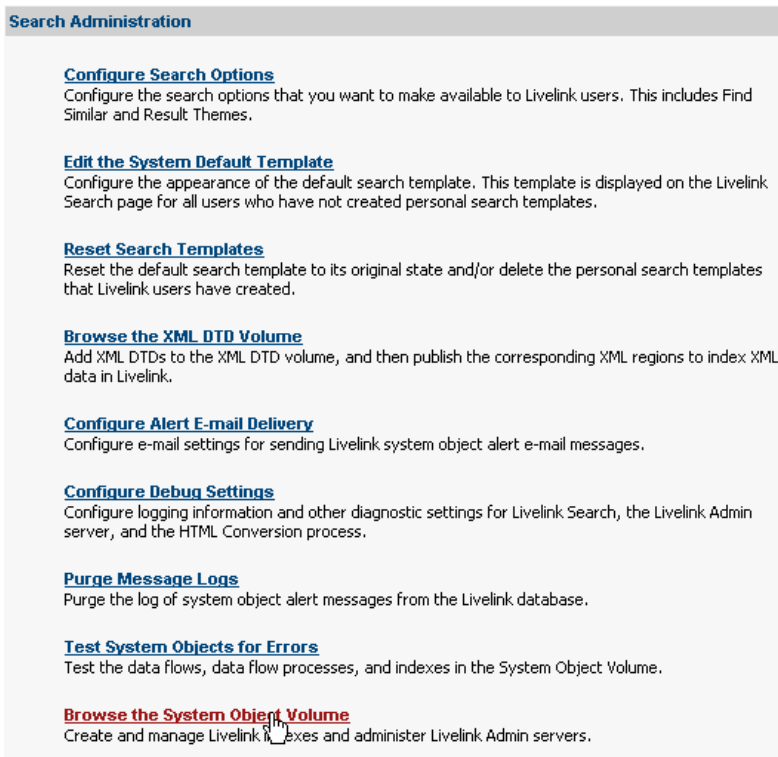


Fig. 50 – The *Search Administration* section on the Livelink Administration page

3. Click the *Browse the System Object Volume* link.
4. Log in with the user information you specified when setting up the Livelink Search server (see section “Installing Livelink Search Servers in the WCM System” on page 173).
5. On the Livelink System page, click *XML Activator Producer Data Source* on the *Add New Item* menu.

General Information	
Slice Replacement:	None...
Processes Prefix:	InternetSite
Port:	8200 The next 6 consecutive port numbers must also be available on all selected hosts.
XML Activator Information	
Host:	Default [WINDOWS NT]
Incoming Directory:	C:\OPENTEXT\index\InternetSite_in
Write Directory:	C:\OPENTEXT\index\InternetSite_write
Operation Tag:	Operation
Identifier Tag:	OTURN
Metadata List:	
Intermediate: HTML Conversion	
Read Directory:	C:\OPENTEXT\index\InternetSite_write
Write Directory:	C:\OPENTEXT\index\InternetSite_write
Host:	Default [WINDOWS NT]
Consumer: Livelink Update Index	
Read Directory:	C:\OPENTEXT\index\InternetSite_write
Index Directory:	C:\OPENTEXT\index\InternetSite_write
Host:	Default [WINDOWS NT]
Start Options	
Start Processes in Data Flow:	<input checked="" type="checkbox"/>
<input type="button" value="Create Processes"/> <input type="button" value="Reset"/>	

Fig. 51 – Creating an XML Activator Producer Data Source

For detailed information on the individual fields, refer to the Livelink online help.

For using the Livelink search in the WCM system, the following fields are of special importance:

- *Process Prefix*: display name for objects associated with this index

When creating a collection, use this identifier as slice name.

- *Incoming Directory*: absolute path to the directory from which the XML Activator Producer process reads the XML files generated by the third-party application

Create this directory manually. If Livelink and the WCM system are installed on different host computers, the directory must be shared.

- *Write Directory*: absolute path to the directory to which the XML Activator Producer process writes the data it extracts

Create this directory on the same file system level as the *Incoming Directory*. The path in this field is automatically entered into the other directory fields of the dialog box.

- *Operation Tag*: Enter "Operation" here.
- *Identifier Tag*: Enter "OTURN" here.

6. Confirm the settings by clicking the *Create Processes* button.

The Livelink system creates a new data source folder. This folder contains all of the system objects that correspond to a particular index: Alias, Data Flow Manager, and Search Manager.

Adding Collections in the WCM System

Note: For detailed information on collections, refer to the Content Miner Manual.

Collections are the basis for searching contents in a WCM-managed website. An internal collection is created from a set of documents that is managed by the WCM system. An external collection is based on a document set that is supplied by any external system, e.g. Livelink.

Adding an internal collection for WCM contents

To add a new internal collection to a Livelink Search server in the WCM system:

1. In the Admin client, select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New internal collection* on the context menu or click the corresponding icon.



Icon for adding an internal collection

3. Use the wizard for adding an internal collection.

For Livelink Search servers, the following dialog box is especially important:

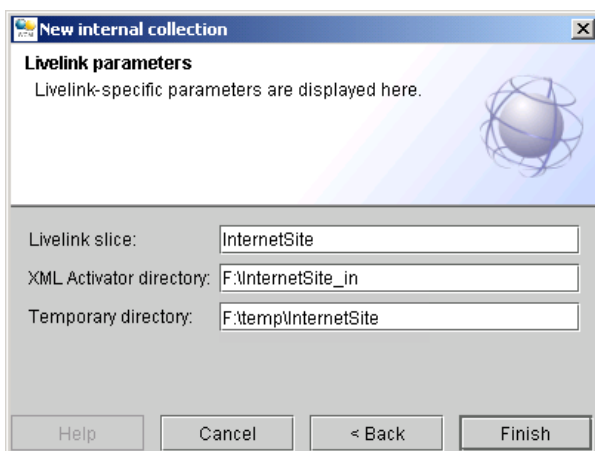


Fig. 52 – Specifying the Livelink parameters for an internal collection

- **Livelink slice:** slice name. Enter the *Processes Prefix* you specified when creating the XML Activator Producer Data Source.

Note: For performance reasons, we recommend that you create only one collection per slice. You can create several internal collections in a slice. In this case, however, you have to make sure that all entries in a slice have a unique URL.

- **XML Activator directory:** The command files created in the temporary directory will be moved to this directory.

Enter the path to the shared *Incoming Directory* that you specified when creating the XML Activator Producer Data Source.

If Livelink and the WCM system are installed on different host computers, the directory must be mapped to a network drive on the WCM system's host computer. Enter the path accordingly.

- *Temporary directory*: The WCM system uses this directory for generating the command files for the XML Activator. In this way, the XML Activator receives instructions when a WCM object changes. The instructions refer to how to process the changes, e.g. re-index the object or remove the object from the index.

Create this directory on the same physical drive as the XML Activator directory.

Before you can use the new collection for search requests, the collection must be synchronized. The synchronization process matches the WCM-managed document set with the document set indexed by the Search server.

4. Select *System administration* → *Search servers* → {*Search server name*} and mark the desired collection.
5. Choose *Synchronize collection* on the context menu or click the corresponding icon.



Icon for synchronizing an internal collection

6. Confirm the security prompts by clicking the *Yes* and *OK* buttons.

Adding an external collection for a Livelink slice

To add an external collection (that is based on a Livelink slice) to the Livelink Search server in the WCM system:

1. Select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New external collection* on the context menu or click the corresponding icon.



Icon for adding an external collection

3. Use the wizard for adding an external collection.

For Livelink Search servers, the following parameter is especially important:

Livelink slice: name of the slice to be indexed

Adding an external collection for another type of document set

To add an external collection (that is based on another document set) to a Livelink Search server in the WCM system:

1. Select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New external collection* on the context menu or click the corresponding icon.



Icon for adding an external collection

3. Use the wizard for adding an external collection.


For Livelink Search servers, the following parameter is especially important:

Livelink slice: slice name. Enter the *Processes Prefix* you specified when creating the XML Activator Producer Data Source.

Note: You can only add one external collection per slice.

4. Before you can use the new collection for search requests, you must add directory structures to the collection by means of the command line tools `AddDocument` and `AddDirectory` (see sections “AddDocument” on page 209 and “AddDirectory” on page 210).


Making Index Regions Queryable in Livelink

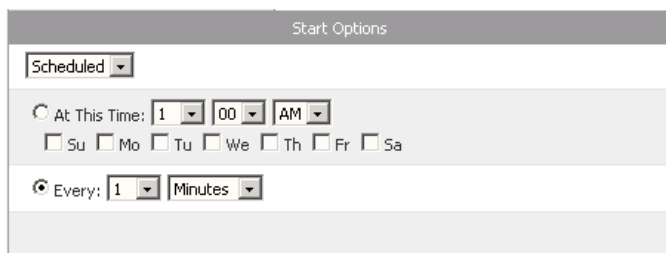
1. On the Livelink System page, click the *{Processes Prefix} Data Source Folder* link.
2. Click the *Functions* icon  of the Search Manager and select *Info* → *Regions*.
3. On the Regions tab, select the *Queryable* check box for all regions starting with the prefix “VIP”.
4. Confirm by clicking the *Update* button.

Starting the XML Activator Process in Livelink

When starting, the XML Activator process scans the specified incoming directory and updates the index according to the XML files located in this directory.

To start the XML Activator process in the Livelink system:

1. On the Livelink System page, click the *{Processes Prefix} Data Source Folder* link.
2. Click the *{Processes Prefix} Data Flow Manager* link.
3. Click the *Functions* icon  of the XML Activator and select *Info* → *Specific*.
4. Make the following settings in the *Start Options* section:



Start Options

Scheduled

☐ At This Time: 1:00 AM


☐ Su ☐ Mo ☐ Tu ☐ We ☐ Th ☐ Fr ☐ Sa

☒ Every: 1 Minutes

Fig. 53 – The *Start Options* section on the *Specific* tab


- Click *Scheduled* in the drop-down list.
- Click the *Every* radio button and enter an interval of 1 minute using the drop-down lists.

Note: If necessary, adapt the interval to the situation in your system.

5. Confirm the entries by clicking the *Update* button.
6. Click the *Functions* icon  of the XML Activator and select *Start*.

Displaying Index Regions in Livelink

After indexing has been completed, you specify in Livelink which index regions will be displayed and can thus be searched via a WCM-managed website.

1. On the Livelink System page, click the *{Processes Prefix} Data Source Folder* link.
2. Click the *Functions* icon  of the Search Manager and select *Info* → *Regions*.
3. To allow users to display a region on their Search Result page, select the region's *Displayable* check box.

Note: If a region does not have a *Displayable* check box, it is because no item with that attribute has yet been added to the Livelink database. After an item with the attribute in question is added and indexed, a *Displayable* check box appears in its row on the Configure Regions page. To add items with this attribute to the database, synchronize the collection. To do so, launch the Admin client and select *System administration* → *Search servers* → *{Search server name}* → *{collection name}*. Right-click the collection and choose *Synchronize collection* on the context menu.

At least the following regions must be marked:

- VIPoid
 - VIPcollection
4. Confirm by clicking the *Update* button.
 5. Choose *Search* on the *Tools* menu.
 6. Click the *Display Options* button.
 7. On the Display Options page, select the *Display* check box for all regions that are to be displayed.
At least the following regions must be marked:
 - VIPoid
 - VIPcollection
 8. Confirm by clicking the *Save* button.

Search Based on Livelink Enterprise Server 9.5.0

Note: Livelink and the WCM system use a shared directory for exchanging data. For this reason, both the Livelink system and the WCM server, to which the Search server is assigned, must have write access to this shared directory.

The Livelink search can be used in different scenarios.

Scenario 1 – Searching website contents

Realizing this scenario requires the following steps in Livelink and the WCM system:

1. WCM system: Install Livelink Search server (see the following section).

2. Livelink system: Create XML Activator Producer data flow (see section “Creating an XML Activator Producer Data Flow in Livelink” on page 193).
3. WCM system: Add collections (see section “Adding an internal collection for WCM contents” on page 197).
4. Livelink system: Start the XML Activator process (see section “Starting the XML Activator Process in Livelink” on page 201).
5. Livelink system: Display index regions (see section “Displaying Index Regions in Livelink” on page 202).

Scenario 2 – Searching Livelink contents

Realizing this scenario requires the following steps in the WCM system:

1. Install Livelink Search server (see the following section).
2. Add collections (see section “Adding an external collection for a Livelink slice” on page 199).

Scenario 3 – Searching other document sets

Realizing this scenario requires the following steps in Livelink and the WCM system:

1. WCM system: Install Livelink Search server (see the following section).
2. Livelink system: Create XML Activator Producer data flow (see section “Creating an XML Activator Producer Data Flow in Livelink” on page 193).
3. WCM system: Add collections (see section “Adding an external collection for another type of document set” on page 200).
4. Livelink system: Start the XML Activator process (see section “Starting the XML Activator Process in Livelink” on page 201).

Installing Livelink Search Servers in the WCM System

Note: For detailed information on installing a WCM system, refer to the Livelink WCM Server Installation Manual.

During the installation of a WCM system, the dialog box *Search server* opens. On the *Livelink* tab of this dialog box, you can add the desired number of Livelink Search servers.

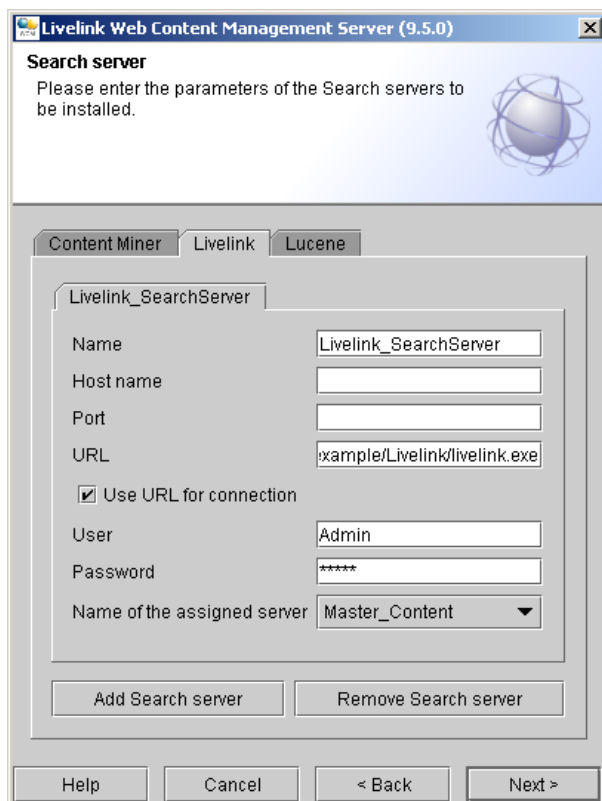


Fig. 54 – Search servers in the WCM installation program

1. Click the *Livelink* tab.

2. Click the *Add Search server* button.

This opens a tab on which you can make the settings for the Livelink Search server.



The screenshot shows the 'Livelink Web Content Management Server (9.5.0)' window. The 'Search server' tab is active, displaying a message: 'Please enter the parameters of the Search servers to be installed.' Below this is a sub-tabbed area with 'Livelink_SearchServer' selected. It contains the following fields and controls:

- Name:** Livelink_SearchServer
- Host name:** (empty text box)
- Port:** (empty text box)
- URL:** xample/Livelink/livelink.exe
- ☒ Use URL for connection
- User:** Admin
- Password:** *****
- Name of the assigned server:** Master_Content (dropdown menu)

At the bottom of the sub-tabbed area are two buttons: 'Add Search server' and 'Remove Search server'. The main window has a 'Help' button and 'Cancel', '< Back', 'Next >' buttons at the very bottom.

Fig. 55 – Installing Livelink Search servers

- **Name:** name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted: a–z, A–Z, 0–9 (ASCII 7 Bit), . (dot), - (hyphen) and _ (underline).

- *Host name*: fully qualified name of the computer hosting the Livelink server which is to perform the indexing and to process the search requests
- *Port*: connection port on the Livelink server (default: 2099)
- *URL*: URL to the Livelink server which is to perform the indexing and to process the search requests. The URL is of type <http://livelinkserver.company.example/Livelink/livelink.exe>.
- *Use URL for connection*: Select this check box if you want to use the Livelink server's URL for the connection. In this case, the entries in the fields *Host name* and *Port* will be ignored.
- *User*: user for processing the search requests in the Livelink system. The user must have read access to the slices.

Notes:

For the Livelink search, you can determine which index regions are displayable (see section “Displaying Index Regions in Livelink” on page 202). These settings are user-specific. For this reason, we recommend that you configure a special user for the Livelink search. Enter the information of this user here.

For further information about permissions for the Livelink search, refer to section “Permissions for Searching” on page 203.

- *Password*: password for processing search requests in the Livelink system
 - *Name of the assigned server*: the name of a WCM server that already exists or is to be installed and to which you want to assign this Search server
3. When you have completed all settings for the Search servers, confirm the entries by clicking the *Next* button.

Creating an XML Activator Producer Data Flow in Livelink

You create an XML Activator Producer data flow to index information generated by a third-party application, such as the WCM system.

To create an XML Activator Producer data flow in the Livelink system:

1. Open the Livelink Administration page.

The URL is of type

<http://livelinkserver.company.example/Livelink/livelink.exe?func=admin.index>.

2. Log in to the Livelink system as the Admin user and navigate to the *Search Administration* section.

Search Administration

[Configure Search Options](#)

Configure the search options that you want to make available to Livelink users. This includes Find Similar and Result Themes.

[Edit the System Default Template](#)

Configure the appearance of the default search template. This template is displayed on the Livelink Search page for all users who have not created personal search templates.

[Reset Search Templates](#)

Reset the default search template to its original state and/or delete the personal search templates that Livelink users have created.

[Open the XML DTD Volume](#)

Add XML DTDs to the XML DTD volume, and then set the corresponding XML regions to allow users to search XML data from Livelink.

[Configure Alert E-mail Delivery](#)

Configure e-mail settings for sending Livelink system object alert e-mail messages.

[Configure Debug Settings](#)

Configure logging information and other diagnostic settings for Livelink Search, the Livelink Admin server, and the HTML Conversion process.

[Purge Message Logs](#)

Purge the log of system object alert messages from the Livelink database.

[Test System Objects for Errors](#)

Test the data flows, data flow processes, and indexes in the System Object Volume.

[Open the System Object Volume](#)

Create and manage Livelink indexes and administer Livelink Admin servers.

Fig. 56 – The Search Administration section on the Livelink Administration page

3. Click the *Open the System Object Volume* link.
4. Log in with the user information you specified when setting up the Livelink Search server (see section “Installing Livelink Search Servers in the WCM System” on page 190).
5. On the Livelink System page, click *XML Activator Producer Data Source* on the *Add Item* menu.

General Information	
Slice Replacement:	None...
Processes Prefix:	InternetSite
Partitions:	1
Port:	8200 The next 12 consecutive port numbers must also be available on all selected hosts.
XML Activator Information	
Host:	Default [WINDOWS NT]
Incoming Directory:	C:\OPENTEXT\index\InternetSite_in
Write Directory:	C:\OPENTEXT\index\InternetSite_write
Operation Tag:	Operation
Identifier Tag:	OTURN
Metadata List:	
Intermediate: Document Conversion	
Read Directory:	C:\OPENTEXT\index\InternetSite_write
Write Directory:	C:\OPENTEXT\index\InternetSite_write
Host:	Default [WINDOWS NT]
Consumer: Livelihood Index	
Read Directory:	C:\OPENTEXT\index\InternetSite_write
Index Directory:	C:\OPENTEXT\index\InternetSite_write
Host:	Default [WINDOWS NT]
Start Options	
Start Processes in Data Flow:	<input checked="" type="checkbox"/>
<div> <div>Create Processes</div> <div>Reset</div> </div>	

Fig. 57 – Creating an XML Activator Producer Data Source

For detailed information on the individual fields, refer to the Livelink online help.

For using the Livelink search in the WCM system, the following fields are of special importance:

- *Process Prefix*: display name for objects associated with this index

When creating a collection, use this identifier as slice name.

- *Incoming Directory*: absolute path to the directory from which the XML Activator Producer process reads the XML files generated by the third-party application

Create this directory manually. If Livelink and the WCM system are installed on different host computers, the directory must be shared.

- *Write Directory*: absolute path to the directory to which the XML Activator Producer process writes the data it extracts

Create this directory on the same file system level as the *Incoming Directory*. The path in this field is automatically entered into the other directory fields of the dialog box.

- *Operation Tag*: Enter “Operation” here.

- *Identifier Tag*: Enter “OTURN” here.

6. Confirm the settings by clicking the *Create Processes* button.

The Livelink system creates a new data source folder. This folder contains all of the system objects that correspond to a particular index: Shortcut, Data Flow Manager, Partition Map, and Search Manager.

Adding Collections in the WCM System

Note: For detailed information on collections, refer to the Content Miner Manual.

Collections are the basis for searching contents in a WCM-managed website. An internal collection is created from a set of documents that is managed by the WCM system. An external collection is based on a document set that is supplied by any external system, e.g. Livelink.

Adding an internal collection for WCM contents

To add a new internal collection to a Livelink Search server in the WCM system:

1. In the Admin client, select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New internal collection* on the context menu or click the corresponding icon.



Icon for adding an internal collection

3. Use the wizard for adding an internal collection.

For Livelink Search servers, the following dialog box is especially important:

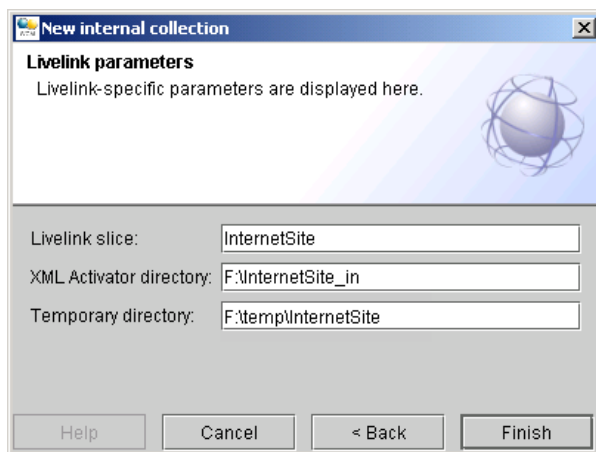


Fig. 58 – Specifying the Livelihood parameters for an internal collection

- *Livelihood slice*: slice name. Enter the *Processes Prefix* you specified when creating the XML Activator Producer Data Source.

Note: For performance reasons, we recommend that you create only one collection per slice. You can create several internal collections in a slice. In this case, however, you have to make sure that all entries in a slice have a unique URL.

- *XML Activator directory*: The command files created in the temporary directory will be moved to this directory.

Enter the path to the shared *Incoming Directory* that you specified when creating the XML Activator Producer Data Source.

If Livelihood and the WCM system are installed on different host computers, the directory must be mapped to a network drive on the WCM system's host computer. Enter the path accordingly.

- *Temporary directory:* The WCM system uses this directory for generating the command files for the XML Activator. In this way, the XML Activator receives instructions when a WCM object changes. The instructions refer to how to process the changes, e.g. re-index the object or remove the object from the index.

Create this directory on the same physical drive as the XML Activator directory.

Before you can use the new collection for search requests, the collection must be synchronized. The synchronization process matches the WCM-managed document set with the document set indexed by the Search server.

4. Select *System administration* → *Search servers* → {*Search server name*} and mark the desired collection.
5. Choose *Synchronize collection* on the context menu or click the corresponding icon.



Icon for synchronizing an internal collection

6. Confirm the security prompts by clicking the *Yes* and *OK* buttons.

Adding an external collection for a Livelink slice

To add an external collection (that is based on a Livelink slice) to the Livelink Search server in the WCM system:

1. Select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New external collection* on the context menu or click the corresponding icon.



Icon for adding an external collection

3. Use the wizard for adding an external collection.

For Livelink Search servers, the following parameter is especially important:

Livelink slice: name of the slice to be indexed

Adding an external collection for another type of document set

To add an external collection (that is based on another document set) to a Livelink Search server in the WCM system:

1. Select *Configuration* → *Search servers* and mark the desired Search server.
2. Choose *New external collection* on the context menu or click the corresponding icon.



Icon for adding an external collection

3. Use the wizard for adding an external collection.

For Livelink Search servers, the following parameter is especially important:

Livelink slice: slice name. Enter the *Processes Prefix* you specified when creating the XML Activator Producer Data Source.


Note: You can only add one external collection per slice.

4. Before you can use the new collection for search requests, you must add directory structures to the collection by means of the command line tools `AddDocument` and `AddDirectory` (see sections “AddDocument” on page 209 and “AddDirectory” on page 210).

Starting the XML Activator Process in Livelink

When starting, the XML Activator process scans the specified incoming directory and updates the index according to the XML files located in this directory.

To start the XML Activator process in the Livelink system:

1. On the Livelink System page, click the *{Processes Prefix} Data Source Folder* link.
2. Click the *{Processes Prefix} Data Flow Manager* link.
3. Click the *Functions* icon  of the XML Activator and select *Properties* → *Specific*.
4. Make the following settings in the *Start Options* section:

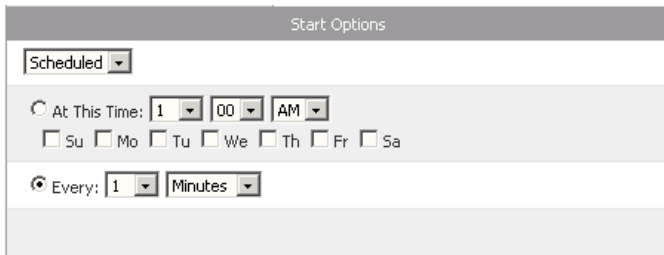




Fig. 59 – The *Start Options* section on the *Specific* tab

- Click *Scheduled* in the drop-down list.
- Click the *Every* radio button and enter an interval of 1 minute using the drop-down lists.

Note: If necessary, adapt the interval to the situation in your system.

5. Confirm the entries by clicking the *Update* button.
6. Click the *Functions* icon  of the XML Activator and select *Start*.

Displaying Index Regions in Livelink

1. On the Livelink System page, click the *{Processes Prefix} Data Source Folder* link.
2. Click the *Functions* icon  of the Search Manager and select *Properties* → *Regions*.
3. Select the *Queryable*, *Displayable*, and *Search By Default* check boxes for all regions starting with the prefix “VIP”.
4. Confirm by clicking the *Update* button.
5. Choose *Search* on the *Tools* menu.
6. Click the *Display Options* button.
7. On the Display Options page, select the *Display* check box for all regions that are to be displayed.

At least the following regions must be marked:

- VIPoid
 - VIPcollection
8. Confirm by clicking the *Save* button.

Permissions for Searching

Each collection is assigned to a Search server. In the Search server's configuration, you specify a Livelink user including password, whose information is used for checking the permissions for search requests in the Livelink system (see section "Installing Livelink Search Servers in the WCM System" on page 173). For formulating such search requests, the `VipContentMinerBean.createQuery()` method is provided.

Scenario 1 – Searching website contents

The collection to be searched is based on an XML Activator Producer Data Source. This data source is a Livelink object for which you can set permissions in the Livelink system. Livelink applies the permissions set for the XML Activator Producer Data Source to all objects contained in the collection.

For a search to return results, the default user specified in the configuration must have read access to the XML Activator Producer Data Source.

The actual access rights are checked by the WCM system. Before displaying search results, the system evaluates the access rights of the logged-in user. The user will only see objects for which the user has read rights in the WCM system.

Scenario 2 – Searching Livelink contents

In the Livelink system, you can set different permissions for each Livelink object contained in the collection. Before displaying search results, the system evaluates the permissions of the default user specified in the configuration. In order to protect confidential documents in your Livelink system from unauthorized access, this user should have the same permissions as the Public Access group.

Alternately, you can specify a different user in search requests. For this purpose, the `VipContentMinerBean.createQuery(username, password)` method is provided. In this case, the permissions of this user and not of the default user will be checked. In this way, you can create personalized search requests.

Scenario 3 – Searching other document sets

The collection to be searched is based on an XML Activator Producer Data Source. This data source is a Livelink object for which you can set permissions in the Livelink system. Livelink applies the permissions set for the XML Activator Producer Data Source to all objects contained in the collection.

For a search to return results, the default user specified in the configuration must have read access to the XML Activator Producer Data Source.

Using the Search

Search in websites

For integrating search functions in a website, the `VipContentMinerBean` is provided. As this bean offers a general search API, you can implement search pages for your website in the same way as for Content Miner or the Search Server Connector for Lucene.

For detailed information on creating search pages, refer to the relevant chapters of the Content Miner Manual or the Search Server Connector for Lucene Manual.

There are two possible ways to define a search for the search engine integrated in Livelink:

1. The *Livelink Query Language* (NLQ) – the syntax of this query language is described in the Livelink User Manual.

2. The Livelink Search API – use a Query object with corresponding additional parameters and a single blank as the search string.

Example:

```
Query query = contentMinerBean.createQuery(serverName);
query.setSearchString(" ");
query.addAdditionalParameter("lookFor1", "allwords");
query.addAdditionalParameter("where1", "bread butter");
query.addAdditionalParameter("boolean1", "AND");
query.addAdditionalParameter("lookFor2", "anywords");
query.addAdditionalParameter("where2", "cheese salad onion");
```

For information on the Livelink Search API, refer to the Using the Search API guide, which is available on the Open Text Knowledge Center (<https://knowledge.opentext.com>).

Search in the Content client

After installing and configuring a Search server, the search functions are also available in the Content client. To learn more about this, refer to the section “Filter and Search Functions” of the Content client User Manual.

As with searching websites, the Livelink Query Language is used here, too.

Command Line Tools

The command line tools are used for managing external collections. They perform the following functions:

- indexing documents for external collections
- retrieving information on external collections and on the Index system used

Note: The command line tools that only read the index (e.g. `GetAvailableCollections`) can also be used for internal collections.

The command line tools are located in the directory **{WCM installation directory}\livelinksearch\{Search server name}\tools**.

To run the command line tools, you must specify appropriate parameters that are required to control the commands. The parameters determine which WCM server is to process the command, and they also contain necessary information about the actual indexing action.

The command parameters that specify the WCM server are identical for all command line tools. For this reason, you can simplify the process of calling the command line tools by defining these default parameters in a configuration file and by specifying this file as a call parameter.

Note: Nearly all command-specific parameters can be specified in the configuration file. We recommend, however, that you include only those parameters in the file that are rarely subject to changes.

The commands.properties Configuration File

By default, the configuration file **commands.properties** is saved with the command line tools in the directory **{WCM installation directory}\livelinksearch\{Search server name}\tools** during the installation. The file contains default parameters corresponding to the information specified during installation. The configuration file looks as follows:

```
#-----  
# configuration properties of the Livelink search server  
# - options in command line have a higher priority  
#-----
```

```
# Incoming directory of the XML Activator Data Source.
xmlActivatorDir = F:\InternetSite_in

# Temporary directory
tempDir = F:\InternetSite_temp

# URL of Livelink CGI
livelinkURL = http://livelinkserver.company.example/livelink/
                                                    livelink.exe

# Username for Livelink connections
username = LivelinkUser

# Password for Livelink connections
password = livelink

# Livelink slice
slice = InternetSite
```

Notes

- The configuration file of the command line tools uses the syntax of Java properties files. Parameters must be specified in the form:
`{Key} = {Value}`
- In contrast to the entries in the command line, the keys (parameter identifiers) in the configuration file are specified without a preceding minus sign (-).
- Backslashes (“\”) are interpreted as escape characters in Java properties files. For this reason, you have to add a backslash before backslashes that really are to be interpreted as such (e.g. in paths).
- Parameters specified in the command line take priority over the corresponding entries in the configuration file.

General Parameters of the Command Line Tools

The parameters described below apply to all commands. They are used to define general settings and the basic behavior of the individual commands. These parameters are mandatory. If a parameter is not specified, the corresponding entry in the **commands.properties** file is used.

Table 10 – General parameters of the command line tools

Parameter	Description
-conf	Configuration File: name of the configuration file. This option can be used to specify the name and/or location of the configuration file. If this parameter is not specified, the file is expected to be in the current directory and to have the name commands.properties . The parameter cannot be specified in the properties file. It is optional.
-livelinkURL	URL: URL to the Livelink server which is to perform the indexing and to process the search requests.
-username	User: user for processing search requests in the Livelink system. The user must have read access to the slices.
-password	Password: password for processing search requests in the Livelink system
-slice	Livelink slice: slice name
-xmlActivatorDir	XML Activator directory: The command files created in the temporary directory will be moved to this directory.
-tempDir	Temporary directory: The WCM system uses this directory for generating the command files for the XML Activator Producer process. In this way, the XML Activator receives instructions when a WCM object changes. The instructions refer to how to process the changes, e.g. re-index the object or remove the object from the index.

AddDocument

AddDocument is used to add a single document specified by its URL to an external collection.

Table 11 – Parameters of the command line tool AddDocument

Parameter	Description
-source	Absolute path in the file system to the document that is to be added
-url	Fully qualified URL of the document that is to be added

Example

```
AddDocument -livelinkURL http://livelinkserver.company.example/livelink/
livelink.exe -username LivelinkUser -password livelink
-slice InternetSite -xmlActivatorDir F:\InternetSite_in
-tempDir F:\InternetSite_temp -source D:\InternetSite\index.html
-url http://wcmsserver.company.example/index.html
```

DeleteDocument

DeleteDocument removes the specified document from an external collection.

Table 12 – Parameters of the command line tool DeleteDocument

Parameter	Description
-url	Fully qualified URL of the document that is to be removed

Example

```
DeleteDocument -livelinkURL http://livelinkserver.company.example/  
livelink/livelink.exe -username LivelinkUser -password livelink  
-slice InternetSite -xmlActivatorDir F:\InternetSite_in  
-tempDir F:\InternetSite_temp  
-url http://wcmserver.company.example/index.html
```

AddDirectory

AddDirectory adds directory structures to an external collection. After the command has been executed, the individual documents of the added directory are part of the index.

Table 13 – Parameters of the command line tool AddDirectory

Parameter	Description
-sourceDir	Absolute path in the file system to the directory that is to be indexed recursively. Subdirectories will be indexed as well.
-baseURL	Fully qualified URL of the directory specified in the parameter -sourceDir. The web server uses this URL to access the directory.

Example

```
AddDirectory -livelinkURL http://livelinkserver.company.example/  
livelink/livelink.exe -username LivelinkUser -password livelink  
-slice InternetSite -xmlActivatorDir F:\InternetSite_in  
-tempDir F:\InternetSite_temp -sourceDir D:\InternetSite  
-baseURL http://wcmserver.company.example/InternetSite
```

DeleteDirectory

DeleteDirectory removes the documents of a directory from the collection. The directory must have been added to the collection using AddDirectory.

Table 14 – Parameters of the command line tool DeleteDirectory

Parameter	Description
-sourceDir	Absolute path in the file system to the directory whose documents are to be removed from the collection. The directory must have been added using AddDirectory.
-baseURL	Fully qualified URL of the directory to be removed from the collection. The web server uses this URL to access the directory.

Example

```
DeleteDirectory -livelinkURL http://livelinkserver.company.example/  
livelink/livelink.exe -username LivelinkUser -password livelink  
-slice InternetSite -xmlActivatorDir F:\InternetSite_in  
-tempDir F:\InternetSite_temp -sourceDir D:\InternetSite  
-baseURL http://wcmserver.company.example/InternetSite
```

GetAvailableCollections

GetAvailableCollections returns a list of all collections contained in the specified slice. The number of documents per collection is also returned.

Example

```
GetAvailableCollections
-livelihoodURL http://livelihoodserver.company.example/livelihood/
livelihood.exe -username LivelihoodUser -password livelihood
-slice InternetSite -xmlActivatorDir F:\InternetSite_in
-tempDir F:\InternetSite_temp
```

ListSources

ListSources returns a list of all documents contained in a collection or in a slice.

Table 15 – Parameters of the command line tool ListSources

Parameter	Description
-collection	Name of the internal collection whose documents are to be listed This parameter or the parameter -all must be specified.
-all	Enter true to list the documents of a slice. This parameter or the parameter -collection must be specified.

Example

```
ListSources -livelinkURL http://livelinkserver.company.example/livelink/
livelink.exe -username LivelinkUser -password livelink
-slice InternetSite -xmlActivatorDir F:\InternetSite_in
-tempDir F:\InternetSite_temp
```

RemoveCollection

RemoveCollection deletes a collection or a slice.

Table 16 – Parameter of the command line tool RemoveCollection

Parameter	Description
-collection	Name of the internal collection that is to be deleted This parameter or the parameter -all must be specified.
-all	Enter true to delete the slice. This parameter or the parameter -collection must be specified.

Example

```
RemoveSources -livelinkURL http://livelinkserver.company.example/
livelink/livelink.exe -username LivelinkUser -password livelink
-slice InternetSite -xmlActivatorDir F:\InternetSite_in
-tempDir F:\InternetSite_temp
```

GetSearchEngineMetadata

GetSearchEngineMetadata returns information on the available index regions of the slice in the Livelink Search server. These regions are defined when configuring the Livelink search (see section “Displaying Index Regions in Livelink” on page 187).

Example

```
GetSearchEngineMetadata
-livelihoodURL http://livelinkserver.company.example/livelink/
livelink.exe -username LivelinkUser -password livelink
-slice InternetSite -xmlActivatorDir F:\InternetSite_in
-tempDir F:\InternetSite_temp
```

Query

Query performs a search in the slice.

Table 17 – Parameters of the command line tool Query

Parameter	Description
-collection	Name of the internal collection that is to be searched This parameter or the parameter -all must be specified.
-all	Enter true to search the slice. This parameter or the parameter -collection must be specified.
-query	String to be found
-batchNr	Batch number. If the batch number is larger than 1, the first n * batch size documents of a search result are skipped.

Parameter	Description
-batchSize	Batch size. Maximum number of documents to be returned as search result.

Problems and Solutions

Problem

The **search** on the Livelink server **does not return any results**, although there are documents in the system matching the search criteria.

Procedure

1. Log in to the Livelink system.

For this purpose, use the URL, user name, and password specified when installing the Search server (see section “Installing Livelink Search Servers in the WCM System” on page 173).

2. In the Search menu, select the slice that you specified when creating the collection to be searched (see section “Adding Collections in the WCM System” on page 182).

3. Enter a search term and select *Go*.

If the Search Results page does not show any results, there are really no appropriate documents in the slice. In this case, there might have been errors during indexing.

4. To check whether the slice contains any documents at all, enter an asterisk * in the search field.
5. If search results are displayed, check whether the fields *VIPcollection* and *VIPoid* are shown.

6. If these two fields are not shown, change the configuration according to the instructions in section “Displaying Index Regions in Livelink” on page 187.
7. Repeat the test.
8. Repeat the test with an extended URL corresponding to the following pattern.
[http://livelinkserver.company.example/livelink/livelink.exe?
"func=search&outputformat=xml&slice={slice name}&nlq=
{search term}&NLQmode=LivelinkAware&hhterms=true](http://livelinkserver.company.example/livelink/livelink.exe?func=search&outputformat=xml&slice={slice name}&nlq={search term}&NLQmode=LivelinkAware&hhterms=true)

Problem

If the **temporary directory** contains many files during normal operation, the files are possibly not moved to the XML Activator directory.

Procedure

1. Check whether the WCM server managing the collection has access to the XML Activator directory and the temporary directory.
2. Check whether it is possible to move files from the temporary directory to the XML Activator directory.

Problem

If the **XML Activator directory** contains many files during normal operation, the XML Activator process might not be running.

Procedure

1. On the Livelink Administration page, select the *Browse the System Object Volume* link in the *Search Administration* section.
2. Select the *{Slice Name} Data Source Folder* link.

3. Select the *{Slice Name} Data Flow Manager* link.
4. Check whether the requests are being processed.
5. Make sure that the slice takes the data from the same directory the collection uses for writing the XML Activator files.

Glossary

API – Application Programming Interface. Livelink WCM Server offers various APIs to access the functionalities of the WCM servers: the WCM Java API, the remote API, the Portal Manager API, and WCM WebServices.

Check in – Staging action of Livelink WCM Server. An object that has been checked out and edited is returned to the WCM system by the action “Check in”. This makes the changes visible in the Edit view. The object is combined with the template again and is available for further editing.

Check out – Staging action of Livelink WCM Server. The content of a WCM object can only be edited after the object has been checked out. A checked-out object is locked for access by other users. The changes to the object do not become available in the Edit view until the object has been checked in.

Collection – An index for a document set is known as a collection. There are internal and external collections. An internal collection is based on a document set managed by a WCM system. An external collection, by contrast, is based on a document set supplied by any external system.

Command line tools – The command line tools can be used to perform indexing actions for external collections. They also make it possible to access information on external and internal collections and on the Index system used.

Data storage view – The data storage view of a server refers to the aspects of the WCM objects that are currently available (Edit view, QA view, Production view). The data storage view available on a server is determined in the routing settings of the website.

Deployment – Deployment is the distribution of data. The deployment of Livelink WCM Server performs two main tasks: first, generating pages from the WCM objects stored in the database and distributing the

generated files to the appropriate directories; second, notifying the WCM servers of changes in the WCM system.

Deployment system – The deployment systems generate pages from the WCM objects and distribute the generated files to the appropriate directories. From there, the files become visible for the users via an HTTP server. Deployment systems may be of various types and categories.

Deployment system category – Depending on the way of processing deployment jobs, deployment systems are assigned to various categories: *Standard deployment systems* automatically generate a new page every time a WCM object is changed. The generated pages are stored in the file system. *Dynamic deployment systems* generate the pages on the basis of user-defined settings and only when the page is requested via the HTTP server. The generated files are stored in a flat file structure. By means of *Search engine deployment systems*, you can prepare your website data for use with a search engine. *WebDAV deployment systems* are required for the use of WebDAV clients. *InSite Editing deployment systems* provide the basis for editing and adding content directly in the website – without the Content client.

Deployment system types – On the basis of the staging concept of Livelink WCM Server, a distinction is made between deployment systems of type “Edit”, “QA”, and “Production”. Different views of the website data are generated, depending on the type.

Dynamization – All or part of the content of a website is not generated until runtime.

Edit view – In the Edit view of Livelink WCM Server, the objects of a website are created and edited. Here the most current status of the objects is visible.

Extranet – Information platform based on Internet technology for business communication with authorized external users, e.g. partners or customers.

Firewall – Hardware or software that monitors the data flow between a public and a private network and protects networks against unauthorized access. Livelink WCM Server supports architectures protected by firewalls.

Group – Collection of users for which specific access rights can be defined. User groups are usually tied to organizational units, such as departments and projects.

Java – Object-oriented programming language developed by Sun Microsystems and used especially in the field of Internet technology. Security aspects and platform independence are the basic philosophies of Java.

JavaScript – Scripting language for integrating executable scripts in HTML pages. JavaScript can be used to incorporate plausibility checks and calculation functions in forms.

JSP – JavaServer-Pages. HTML files with specifically identified embedded Java programs that are converted into servlets by using the JSP engine and then executed on the web server. The result is then sent to the client in normal HTML format (without Java).

LDAP – Lightweight Directory Access Protocol. The LDAP is based on the X.500 standard and is supported by most major software manufacturers. LDAP directory services are used to manage user information.

Master server – Only master servers have read and write access to the data of a WCM system. The master Content server manages website data, while the master Administration server manages the configuration and system data of the WCM system. See also *Server category*.

Metadata – Every WCM object has a number of object information assigned to it (e.g. expiration date, language). These are known as metadata.

Object – Each element of a website is integrated in Livelink WCM Server as a single object. Each object is based on an object type, such as “HTML

page". Object types can be defined in the Admin client or the Content client.

Object type – The specific kind of object, e.g. "HTML page", "HTML template", "Topic". Various properties of the WCM object result from the object type. The object type is defined when the object is created. There are only a few cases in which it may subsequently be changed. Object types can be edited in the Admin client or the Content client.

Pool – The different connections within a WCM system are managed in pools. These include, for example, connections for communication with an LDAP directory service or for communication between the WCM servers. If a connection is required, it is taken from the respective pool. After the data transfer, the connection is returned to the pool. Pools always combine connections of the same type, e.g. connections to databases (JDBC pools) or connections between WCM servers (WCM pools).

Portal – A portal is a website that serves the user as a central point of access – as a gate – to certain Internet services. A portal often offers topic-specific and personalized offers and information.

Production view – The Production view of Livelink WCM Server makes the released pages of a website available to the user. By means of a web server, these pages can be accessed in the Internet, intranet, or extranet.

Properties file – A file containing resource information in a defined format (key-value pairs)

Proxy server – A proxy server is used to intercept requests from a client application, e.g. a browser, to one or more other servers. If the proxy server can meet the request, it sends the requested data back to the client. Otherwise, it forwards the request to the specified server.

In the context of Livelink WCM Server, WCM servers of the category "proxy" do not have write access, but only read access to the WCM objects or the configuration. Changes to the WCM objects are only possible via the master Content server, changes to the configuration of the

WCM system are made only via the master Administration server. See also *Server category*.

QA view – The QA view of Livelink WCM Server is used for quality assurance of the objects and thus of the website content. This view thus performs the control function between editing in the Edit view and publication in the Production view.

Query system – The Query system provides a mechanism for processing search requests and generating corresponding hitlists.

Relator – Due to the integration of web content management and Livelink, you can add Livelink objects as WCM objects in a WCM-managed website. These WCM objects are called relators. They always refer to a certain version of a Livelink object. For integrating a single Livelink object, the WCM system provides the object type “Livelink relator”. For integrating Livelink folders, the WCM object type “Livelink folder relator” is provided.

Release – Staging action of Livelink WCM Server. Quality Assurance checks whether the content and the form of a submitted object meet the quality standards of the company. If so, the object is released. The release transmits the quality-assured version of the object to the Production view, and thus makes it available to the end user in the published website.

Repository – A repository is an abstraction that allows uniform access to data.

Role – Collection of users, similar to a user group, for which specific access rights can be defined. The user role is usually defined in terms of tasks, whereas user groups are generally tied to organizational units, such as departments or projects.

Search server – The combination of Index and Query system in Livelink WCM Server is called “Search server”. Each Search server is assigned to exactly one WCM server. It is, however, possible to assign more than one Search server to a WCM server.

Server category – In a WCM system, a distinction is made between master and proxy servers. Master servers have write access to the data of the WCM system, while proxy servers have only read access. The master Content server manages the website data, the master Administration server manages the configuration and system data. In addition to this, any number of proxy servers can be set up.

Server type – According to the tasks of the servers, there are two server types: Content servers for managing website data and Administration servers for managing the user, configuration, and system data of the WCM system. Basically, every Content server is able to provide all views of the data of the managed websites – Edit, QA, and Production. The available views may be limited by the fact that the Content server only receives the data of certain views.

Status – The processing state of a WCM object. Changes to the state are caused by the corresponding actions on the WCM object.

Submit – Staging action of Livelink WCM Server. before a newly created or changed object can be published, it must be submitted to Quality Assurance for checking. This makes the changes to the object visible in the QA view.

Synchronization – The synchronization process matches the WCM-managed document set with the document set indexed by the Search server. A completely new index is generated.

Template – File that provides the framework for other objects. The HTML pages for the Production view are created by bringing together the template and the content. Any HTML components may be integrated in a template.

Topic – Combination of an HTML page for direct display of information and a list for accommodating subordinate objects. The topics are shown as nodes in the object tree and thus help structure the website.

URL – Uniform Resource Locator. A unique address in the World Wide Web.

WCM server – In a WCM system, there are several WCM servers working in parallel (server processes). The exact tasks of a server depend on the server type and server category.

WCM tag – Special element for integrating WCM specific data in an HTML page. WCM tags are used in particular when creating templates.

Index

A

- access to Admin client
 - groups/roles 45
 - users 37
 - World 47
- add
 - external collection (Enterprise Server 9.2.1) 184
 - external collection (Enterprise Server 9.5.0) 199
 - folder relator 139
 - functional area 49
 - group (Admin client) 42
 - group (Livelihood) 73
 - group to group (Livelihood) 75
 - internal collection (Enterprise Server 9.2.1) 182
 - internal collection (Enterprise Server 9.5.0) 197
 - Livelihood object as WCM object 131
 - relator 134
 - relator with Livelihood document 144
 - role (Admin client) 42
 - user (Admin client) 34
 - user (Livelihood) 61
 - user to group (Livelihood) 75
- AddDirectory 210
- AddDocument 209
- Admin client
 - filter functions 31
- administration rights
 - groups/roles 45
 - users 37
 - World 47
- API
 - Portal Manager API 91
 - WCM Java API 92

- application
 - search websites 204
- application example
 - channel in website 108
 - discussion as FAQ 120
 - discussion in website 113
 - integrate folder content in a website 95
 - Livelihood browser 100
 - poll on website 123
- application examples 95
- assign
 - functional areas to principals 51
 - websites to principals 52
- assignments
 - functional areas 51
 - groups (Admin client) 43
 - roles (Admin client) 43
 - users (Admin client) 37
 - websites 52
- automatic synchronization 82

B

- beans for Livelihood integration 91

C

- channel 108
 - integrate in website (application example) 108
- collection
 - add external (Enterprise Server 9.2.1) 184
 - add external (Enterprise Server 9.5.0) 199
 - add internal (Enterprise Server 9.2.1) 182

- add internal (Enterprise Server 9.5.0) 197
- external 182, 197
- internal 182, 197
- command line tools
 - AddDirectory 210
 - AddDocument 209
 - configuration file 206
 - DeleteDirectory 211
 - DeleteDocument 209
 - general parameters 208
 - GetAvailableCollections 212
 - GetSearchEngineMetadata 214
 - ListSources 212
 - Query 214
 - RemoveCollection 213
- commands.properties 206
- common user administration
 - in general 19
 - installation 19
- configuration
 - multiple Livelink systems 86
- Content client search 205

D

- delete
 - functional area 52
 - group (Admin client) 46
 - group (Livelink) 79
 - relator 163
 - role (Admin client) 46
 - user (Admin client) 39
 - user (Livelink) 71
- DeleteDirectory 211
- DeleteDocument 209
- discussion
 - integrate in website (application example) 113
 - provide as FAQ (application example) 120

E

- edit
 - content of a relator 154
 - Livelink metadata 155
- external collection 182, 197
 - add (Enterprise Server 9.2.1) 184
 - add (Enterprise Server 9.5.0) 199

F

- FAQ
 - application example 120
- file.xml.tok 177
- filter functions
 - Admin client 31
 - Livelink 57
- find
 - changed relator 150
- Fixed version
 - folder relator 143
 - relator 138
- folder relator 131
 - add 139
- functional areas
 - assign 51
 - delete 52
 - manage 47
 - new 49
 - settings 50
- functions
 - user administration (Admin client) 30

G

- GetAvailableCollections 212
- GetSearchEngineMetadata 214
- group leader
 - set (Livelink) 77
 - unset (Livelink) 78

groups

- add (Livelink) 73
- add group (Livelink) 75
- add user (Livelink) 75
- administration rights 45
- assign functional areas 51
- assign websites 52
- assignments (Admin client) 43
- change name (Livelink) 78
- delete (Admin client) 46
- delete (Livelink) 79
- manage (Admin client) 40
- manage (Livelink) 72
- new (Admin client) 42
- remove group (Livelink) 79
- remove user (Livelink) 79
- set as group leader (Livelink) 77
- set group leader (Livelink) 77
- settings (Admin client) 43
- settings (Livelink) 74
- unset group leader (Livelink) 78

I**index regions**

- display 188, 202
- for the WCMS 177
- make displayable (Enterprise Server 9.2.1) 187
- make queryable (Enterprise Server 9.2.1) 186
- make queryable (Enterprise Server 9.5.0) 202

installation

- automatic synchronization 82
- common user administration 19
- Livelink Search server (Enterprise Server 9.2.1) 173
- Livelink Search server (Enterprise Server 9.5.0) 190
- WCM Kernel Integration module 82

internal collection 182, 197

- add (Enterprise Server 9.2.1) 182
- add (Enterprise Server 9.5.0) 197
- synchronize 184, 199

introduction 11**L****ListSources 212****Livelink**

- common user administration 19
- search functions 57

Livelink browser (application example) 100**Livelink document**

- add with relator 144

Livelink Enterprise Server 9.2.1

- set up search 171

Livelink Enterprise Server 9.5.0

- set up search 188

Livelink folder

- integrate content in website (application example) 95

Livelink folder relator 131

- add 139

Livelink functions in a website 81**Livelink ID 137****Livelink metadata**

- edit 155
- integrate in page 159
- WCM tags 159
- WCM tags for standard metadata 159
- WCM tags for user-defined metadata 160

Livelink metadata dialog box 155**Livelink module**

- WCM Kernel Integration 82
- WCM User Administration 20

Livelink object

- add as WCM object 131

Livelink relator 131

- add 134
- add with document 144
- configuring automatic synchronization 82
- delete 163
- edit content 154
- edit metadata 155
- find after change 150
- integrate metadata 159
- release 153
- submit 152
- update manually 162

Livelink search

- add collections (Enterprise Server 9.2.1) 182
- add collections (Enterprise Server 9.5.0) 197
- index regions (Enterprise Server 9.2.1) 186
- index regions (Enterprise Server 9.5.0) 202
- install Search server (Enterprise Server 9.2.1) 173
- install Search server (Enterprise Server 9.5.0) 190
- permissions 203
- prepare indexing (Enterprise Server 9.2.1) 177
- set up 171
- XML Activator (Enterprise Server 9.2.1) 178
- XML Activator (Enterprise Server 9.5.0) 193

Livelink Search server

- install (Enterprise Server 9.2.1) 173
- install (Enterprise Server 9.5.0) 190

Livelink system

- configure multiple 86

LivelinkObjectBean 91

LivelinkUserBean 91

M

manage

- functional areas 47
- groups (Admin client) 40
- groups (Livelink) 72
- roles (Admin client) 40
- users (Admin client) 33
- users (Livelink) 60

module

- WCM Kernel Integration 82
- WCM User Administration 20

multiple Livelink systems 86

N

new

- external collection (Enterprise Server 9.2.1) 184
- external collection (Enterprise Server 9.5.0) 199
- folder relator 139
- functional area 49
- group (Admin client) 42
- group (Livelink) 73
- internal collection (Enterprise Server 9.2.1) 182
- internal collection (Enterprise Server 9.5.0) 197
- Livelink object as WCM object 131
- relator 134
- relator with Livelink document 144
- role (Admin client) 42
- user (Admin client) 34
- user (Livelink) 61

news 109

P

permissions

- for searching 203
- in Livelink 94

- poll on website (application example) 123
- Portal Manager API 91
 - beans for Livelink integration 91
- preparations for using Livelink functions 82
- principals
 - find (Admin client) 31
 - find (Livelink) 57
 - manage (Admin client) 26
 - manage (Livelink) 54

Q

- Query 214

R

- relator 131
 - add 134
 - add with Livelink document 144
 - configuring automatic synchronization 82
 - delete 163
 - edit content 154
 - edit metadata 155
 - find after change 150
 - integrate metadata 159
 - release 153
 - submit 152
 - update manually 162
- release
 - relator 153
- remove
 - functional area 52
 - group (Admin client) 46
 - group (Livelink) 79
 - group from group (Livelink) 79
 - role (Admin client) 46
 - user (Livelink) 71
 - user from group (Livelink) 79
 - users (Admin client) 39

- RemoveCollection 213
- roles
 - administration rights 45
 - assign functional areas 51
 - assign websites 52
 - assignments (Admin client) 43
 - delete (Admin client) 46
 - manage (Admin client) 40
 - new (Admin client) 42
 - settings (Admin client) 43

S

- search
 - add collections (Enterprise Server 9.2.1) 182
 - add collections (Enterprise Server 9.5.0) 197
 - in the Content client 205
 - in websites 204
 - index regions (Enterprise Server 9.2.1) 186
 - index regions (Enterprise Server 9.5.0) 202
 - install Search server (Enterprise Server 9.2.1) 173
 - install Search server (Enterprise Server 9.5.0) 190
 - Livelink search for the WCMS 171
 - permissions 203
 - prepare indexing (Enterprise Server 9.2.1) 177
 - principals (Admin client) 31
 - principals (Livelink) 57
 - with Livelink Enterprise Server 9.2.1 171
 - with Livelink Enterprise Server 9.5.0 188
 - XML Activator (Enterprise Server 9.2.1) 178
 - XML Activator (Enterprise Server 9.5.0) 193

- set up
 - automatic synchronization 82
 - common user administration 19
 - Livelink search 171
 - search with Livelink Enterprise Server 9.2.1 171
 - search with Livelink Enterprise Server 9.5.0 188
- settings
 - edit user information (Admin client) 36
 - edit user information (Livelink) 64
 - functional areas 50
 - group (Admin client) 43
 - group (Livelink) 74
 - role (Admin client) 43
- submit
 - relator 152
- T**
 - tab General
 - edit 64
 - tab Personal
 - edit 64
 - tab WCM
 - edit 64
 - temporary directory 184, 199
- U**
 - update
 - relator manually 162
 - Update Edit version automatically
 - folder relator 142
 - relator 137
 - user administration
 - common 19
 - functions (Admin client) 30
 - in Livelink 54
 - in the Admin client 26
 - install module 20
 - Livelink 19
 - set up common 19
 - user profile
 - modify 64
 - user profile (Livelink) 64
 - users
 - administration rights 38
 - assign functional areas 51
 - assign websites 52
 - assignments (Admin client) 37
 - delete (Admin client) 39
 - delete (Livelink) 71
 - manage (Admin client) 33
 - manage (Livelink) 60
 - modify settings (Admin client) 36
 - modify settings (Livelink) 64
 - new (Admin client) 34
 - new (Livelink) 61
 - set group leader (Livelink) 77
 - unset group leader (Livelink) 78
 - using Livelink functions
 - preparations 82
- W**
 - WCM Java API 92
 - WCM Kernel Integration
 - install 82
 - install module 82
 - uninstall 83
 - WCM tags
 - for Livelink standard attributes 159
 - for Livelink standard metadata 159
 - for user-defined attributes 160
 - for user-defined metadata 160
 - WCM User Administration
 - install 20
 - uninstall 21
 - websites
 - assign 52

World

administration rights 47

X

XML Activator

create producer data flow

(Enterprise Server 9.2.1) 178

create producer data flow

(Enterprise Server 9.5.0) 193

directory 183, 198

processes 168

xmlactiv 177

